(12) **United States Patent**
Bansod et al.

(10) **Patent No.: US 11,366,723 B2**
(45) **Date of Patent: Jun. 21, 2022**

(54) **DATA STORAGE MANAGEMENT SYSTEM FOR HOLISTIC PROTECTION AND MIGRATION OF SERVERLESS APPLICATIONS ACROSS MULTI-CLOUD COMPUTING ENVIRONMENTS**

(71) Applicant: **Commvault Systems, Inc.**, Tinton Falls, NJ (US)

(72) Inventors: **Ketan Bansod**, Morganville, NJ (US); **Jun H. Ahn**, Manalapan, NJ (US); **Girish Ramohalli Gopala Rao**, Eatontown, NJ (US)

(73) Assignee: **Commvault Systems, Inc.**, Tinton Falls, NJ (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 138 days.

(21) Appl. No.: **16/709,668**

(22) Filed: **Dec. 10, 2019**

(65) **Prior Publication Data**

US 2020/0351345 A1 Nov. 5, 2020

Related U.S. Application Data

(60) Provisional application No. 62/841,126, filed on Apr. 30, 2019.

(51) **Int. Cl.**

G06F 11/14 (2006.01)
H04L 29/08 (2006.01)
(Continued)

(52) **U.S. Cl.**

CPC ***G06F 11/1464*** (2013.01); ***G06F 11/1451*** (2013.01); ***G06F 11/1469*** (2013.01);
(Continued)

(58) **Field of Classification Search**

CPC G06F 11/1451; G06F 11/1464; G06F 11/1469; G06F 11/203; G06F 2201/84; H04L 67/1095; H04L 67/16; H04L 67/34
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,686,620	A	8/1987	Ng
4,995,035	A	2/1991	Cole et al.

(Continued)

FOREIGN PATENT DOCUMENTS

EP	0259912	3/1988
EP	0405926	1/1991

(Continued)

OTHER PUBLICATIONS

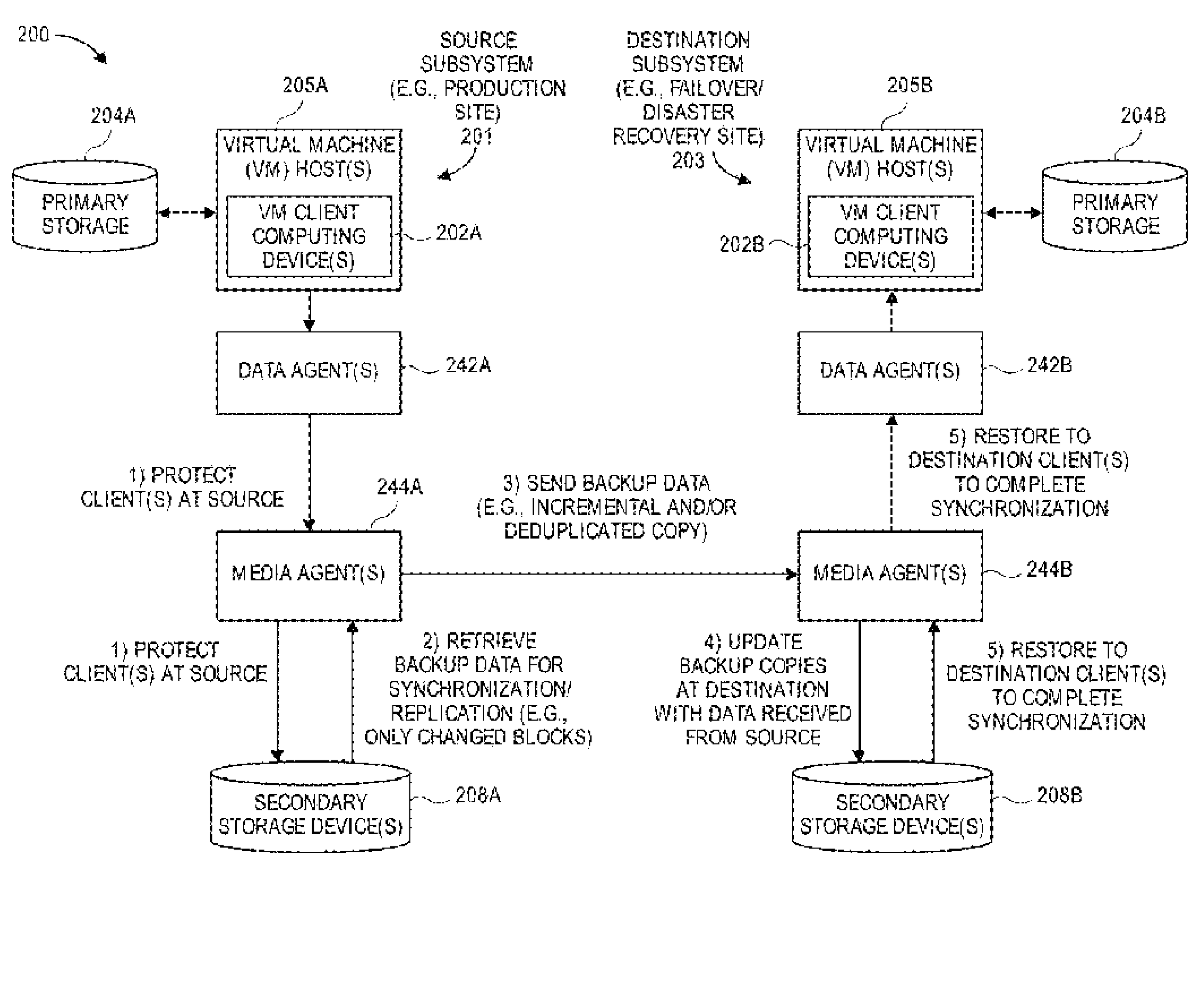
U.S. Appl. No. 16/721,644, filed Dec. 19, 2019, Kumar et al.
(Continued)

Primary Examiner — Thomas J Dailey
(74) *Attorney, Agent, or Firm* — Knobbe, Martens, Olson & Bear, LLP

(57) **ABSTRACT**

The present inventors devised a holistic approach for protecting serverless applications in single-cloud, multi-zone, multi-cloud, and/or non-cloud data center computing environments. An illustrative data storage management system discovers application assets, relationships, and interoperability dependencies and creates an "application entity" that references the various assets. Protection preferences apply to the application entity as a whole. An orchestration function in the system coordinates storage management operations (e.g., backup, replication, live synchronization, etc.) in a suitable order of operations gleaned from asset dependencies, if any. A set of copies of the application's discovered assets are generated in coordinated fashion and represent a point-in-time copy of the application. The point-in-time copy can be restored and/or migrated to other computing services by the data storage management system. The orchestration function coordinates restore and migration operations, including any cloud-to-cloud or cloud-to/from-non-cloud conversions that might be necessary to activate the application in a different computing environment.

20 Claims, 28 Drawing Sheets



(51) **Int. Cl.**

G06F 11/20	(2006.01)
H04L 67/1095	(2022.01)
H04L 67/1097	(2022.01)
H04L 67/51	(2022.01)
H04L 67/00	(2022.01)

(52) **U.S. Cl.**
CPC ***G06F 11/203*** (2013.01); ***H04L 67/1095*** (2013.01); ***H04L 67/1097*** (2013.01); ***H04L 67/16*** (2013.01); ***H04L 67/34*** (2013.01); *G06F 2201/84* (2013.01)

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,005,122 A 4/1991 Griffin et al.
5,093,912 A 3/1992 Dong et al.
5,123,107 A 6/1992 Mensch, Jr.
5,133,065 A 7/1992 Cheffetz et al.
5,193,154 A 3/1993 Kitajima et al.
5,212,772 A 5/1993 Masters
5,226,157 A 7/1993 Nakano et al.
5,239,647 A 8/1993 Anglin et al.
5,241,668 A 8/1993 Eastridge et al.
5,241,670 A 8/1993 Eastridge et al.
5,276,860 A 1/1994 Fortier et al.
5,276,867 A 1/1994 Kenley et al.
5,287,500 A 2/1994 Stoppani, Jr.
5,321,816 A 6/1994 Rogan et al.
5,333,315 A 7/1994 Saether et al.
5,347,653 A 9/1994 Flynn et al.
5,410,700 A 4/1995 Fecteau et al.
5,448,724 A 9/1995 Hayashi et al.
5,491,810 A 2/1996 Allen
5,495,607 A 2/1996 Pisello et al.
5,504,873 A 4/1996 Martin et al.
5,544,345 A 8/1996 Carpenter et al.
5,544,347 A 8/1996 Yanai et al.
5,559,957 A 9/1996 Balk
5,619,644 A 4/1997 Crockett et al.
5,638,509 A 6/1997 Dunphy et al.
5,664,204 A 9/1997 Wang
5,673,381 A 9/1997 Huai et al.
5,699,361 A 12/1997 Ding et al.
5,729,743 A 3/1998 Squibb
5,751,997 A 5/1998 Kullick et al.
5,758,359 A 5/1998 Saxon
5,761,677 A 6/1998 Senator et al.
5,764,972 A 6/1998 Crouse et al.
5,778,395 A 7/1998 Whiting et al.
5,812,398 A 9/1998 Nielsen
5,813,009 A 9/1998 Johnson et al.
5,813,017 A 9/1998 Morris
5,875,478 A 2/1999 Blumenau
5,887,134 A 3/1999 Ebrahim
5,901,327 A 5/1999 Ofek
5,924,102 A 7/1999 Perks
5,950,205 A 9/1999 Aviani, Jr.
5,974,563 A 10/1999 Beeler, Jr.
6,021,415 A 2/2000 Cannon et al.
6,026,414 A 2/2000 Anglin
6,052,735 A 4/2000 Ulrich et al.
6,076,148 A 6/2000 Kedem et al.
6,094,416 A 7/2000 Ying
6,131,095 A 10/2000 Low et al.
6,131,190 A 10/2000 Sidwell
6,148,412 A 11/2000 Cannon et al.
6,154,787 A 11/2000 Urevig et al.
6,161,111 A 12/2000 Mutalik et al.
6,167,402 A 12/2000 Yeager
6,212,512 B1 4/2001 Barney et al.
6,260,069 B1 7/2001 Anglin
6,269,431 B1 7/2001 Dunham
6,275,953 B1 8/2001 Vahalia et al.
6,301,592 B1 10/2001 Aoyama et al.
6,324,581 B1 11/2001 Xu et al.
6,327,590 B1 12/2001 Chidlovskii et al.
6,328,766 B1 12/2001 Long
6,330,570 B1 12/2001 Crighton et al.
6,330,642 B1 12/2001 Carteau
6,343,324 B1 1/2002 Hubis et al.
RE37,601 E 3/2002 Eastridge et al.
6,356,801 B1 3/2002 Goodman et al.
6,389,432 B1 5/2002 Pothapragada et al.
6,421,711 B1 7/2002 Blumenau et al.
6,487,561 B1 11/2002 Ofek et al.
6,519,679 B2 2/2003 Devireddy et al.
6,538,669 B1 3/2003 Lagueux, Jr. et al.
6,564,228 B1 5/2003 O'Connor
6,658,526 B2 12/2003 Nguyen et al.
6,941,429 B1 9/2005 Kamvyssells et al.
6,959,327 B1 10/2005 Vogl
6,973,555 B2 12/2005 Fujiwara
7,000,238 B2 2/2006 Nadler
7,035,880 B1 4/2006 Crescenti
7,079,341 B2 7/2006 Kistler et al.
7,096,418 B1 8/2006 Singhal
7,107,298 B2 9/2006 Prahlad
7,130,272 B1 10/2006 Gai et al.
7,143,203 B1 11/2006 Altmejd
7,225,220 B2 5/2007 Gonzalez et al.
7,246,207 B2 7/2007 Kottomtharayil
7,260,633 B2 8/2007 Lette
7,315,923 B2 1/2008 Retnamma et al.
7,334,144 B1 2/2008 Schlumberger
7,340,616 B2 3/2008 Rothman et al.
7,343,356 B2 3/2008 Prahlad
7,343,453 B2 3/2008 Prahlad
7,346,751 B2 3/2008 Prahlad
7,366,846 B2 4/2008 Boyd et al.
7,386,744 B2 6/2008 Barr
7,395,282 B1 7/2008 Crescenti
7,448,079 B2 11/2008 Tremain
7,472,079 B2 12/2008 Fellenstein
7,483,895 B2 1/2009 Hysom
7,502,820 B2 3/2009 Manders
7,516,346 B2 4/2009 Pinheiro et al.
7,516,348 B1 4/2009 Ofer
7,526,798 B2 4/2009 Chao
7,529,782 B2 5/2009 Prahlad et al.
7,546,475 B2 6/2009 Mayo et al.
7,587,570 B2 9/2009 Sarkar et al.
7,617,191 B2 11/2009 Wilbrink et al.
7,617,262 B2 11/2009 Prahlad et al.
7,627,827 B2 12/2009 Taylor et al.
7,631,351 B2 12/2009 Erofeev
7,651,593 B2 1/2010 Prahlad
7,653,668 B1 1/2010 Shelat
7,668,884 B2 2/2010 Prahlad
7,685,269 B1 3/2010 Thrasher et al.
7,694,070 B2 4/2010 Mogi
7,734,669 B2 6/2010 Kottomtharayil et al.
7,739,548 B2 6/2010 Goodrum et al.
7,747,579 B2 6/2010 Prahlad et al.
7,761,736 B2 7/2010 Nguyen et al.
7,765,167 B2 7/2010 Prahlad
7,769,616 B2 8/2010 Ollivier
7,778,984 B2 8/2010 Zhang
7,792,789 B2 9/2010 Prahlad
7,797,453 B2 9/2010 Meier et al.
7,814,149 B1 10/2010 Stringham
7,814,351 B2 10/2010 Redlich et al.
7,818,082 B2 10/2010 Roumeliotis et al.
7,822,967 B2 10/2010 Fung
7,840,537 B2 11/2010 Gokhale
7,882,077 B2 2/2011 Gokhale
7,899,788 B2 3/2011 Chadhok et al.
7,917,438 B2 3/2011 Kenedy et al.
7,975,061 B1 7/2011 Gokhale et al.
7,996,270 B2 8/2011 Sundaresan
8,001,277 B2 8/2011 Mega
8,037,028 B2 10/2011 Prahlad
8,065,166 B2 11/2011 Maresh
8,108,427 B2 1/2012 Prahlad
8,112,605 B2 2/2012 Kavuri

(56) **References Cited**

U.S. PATENT DOCUMENTS

8,134,727	B1	3/2012	Shmunis
8,140,786	B2	3/2012	Bunte
8,156,086	B2	4/2012	Lu et al.
8,170,995	B2	5/2012	Prahlad et al.
8,219,524	B2	7/2012	Gokhale
8,230,195	B2	7/2012	Amarendran et al.
8,266,406	B2	9/2012	Kavuri
8,285,681	B2	10/2012	Prahlad
8,296,534	B1	10/2012	Gupta et al.
8,307,177	B2	11/2012	Prahlad
8,316,091	B2	11/2012	Hirvela et al.
8,321,688	B2	11/2012	Auradkar
8,352,608	B1	1/2013	Keagy et al.
8,364,652	B2	1/2013	Vijayan et al.
8,364,802	B1	1/2013	Keagy et al.
8,370,307	B2	2/2013	Wolfe
8,396,838	B2	3/2013	Brockway et al.
8,407,190	B2	3/2013	Prahlad
8,417,697	B2	4/2013	Ghemawat et al.
8,429,630	B2	4/2013	Nickolov
8,434,131	B2	4/2013	Varadharajan et al.
8,510,573	B2	8/2013	Muller
8,527,549	B2	9/2013	Cidon
8,566,362	B2	10/2013	Mason et al.
8,578,120	B2	11/2013	Attarde et al.
8,612,439	B2	12/2013	Prahlad
8,626,741	B2	1/2014	Vijakumar et al.
8,635,184	B2	1/2014	Hsu et al.
8,660,038	B1	2/2014	Pascazio
8,674,823	B1	3/2014	Contrario et al.
8,683,103	B2	3/2014	Ripberger
8,707,070	B2	4/2014	Muller
8,769,048	B2	7/2014	Kottomtharayil
8,780,400	B2	7/2014	Shmunis
8,799,242	B2	8/2014	Leonard et al.
8,849,761	B2	9/2014	Prahlad
8,849,955	B2	9/2014	Prahlad
8,924,511	B2	12/2014	Brand
8,938,481	B2	1/2015	Kumarasamy et al.
8,950,009	B2	2/2015	Vijayan et al.
8,954,446	B2	2/2015	Vijayan et al.
9,020,900	B2	4/2015	Vijayan et al.
9,021,282	B2	4/2015	Muller et al.
9,021,307	B1	4/2015	Parameswaran et al.
9,098,495	B2	8/2015	Gokhale
9,176,773	B2 *	11/2015	Fries G06F 9/4856
9,189,170	B2	11/2015	Kripalani et al.
9,195,636	B2	11/2015	Smith
9,239,687	B2	1/2016	Vijayan et al.
9,262,496	B2	2/2016	Kumarasamy et al.
9,307,177	B2	4/2016	Park et al.
9,336,226	B2	5/2016	Vibhor et al.
9,436,555	B2	9/2016	Dornemann et al.
9,444,811	B2	9/2016	Nara et al.
9,454,537	B2	9/2016	Prahlad et al.
9,633,033	B2	4/2017	Vijayan et al.
9,641,388	B2	5/2017	Kripalani et al.
9,648,100	B2	5/2017	Klose et al.
9,766,825	B2	9/2017	Bhagi et al.
9,852,026	B2	12/2017	Mitkar et al.
9,959,333	B2	5/2018	Kumarasamy
10,228,962	B2	3/2019	Dornemann et al.
10,248,657	B2	4/2019	Prahlad et al.
10,255,143	B2	4/2019	Vijayan et al.
10,264,074	B2	4/2019	Vijayan et al.
10,296,368	B2	5/2019	Dormemann et al.
10,346,259	B2	7/2019	Gokhale et al.
10,379,598	B2	8/2019	Muller
10,387,073	B2	8/2019	Bhagi et al.
10,387,266	B2	8/2019	Kumarasamy et al.
10,402,090	B1	9/2019	Ssaur et al.
10,592,145	B2	3/2020	Bedadala et al.
10,592,350	B2	3/2020	Dornemann
10,649,861	B1	5/2020	Natanzon et al.
10,664,352	B2	5/2020	Rana
10,684,924	B2	6/2020	Kilaru et al.
10,740,193	B2	8/2020	Dhatrak
10,749,943	B1	8/2020	Feiguine et al.
10,768,971	B2	9/2020	Dornemann et al.
10,776,329	B2	9/2020	Rao et al.
10,782,934	B1 *	9/2020	Chawda G06F 9/5088
10,936,435	B2	3/2021	Natanzon et al.
11,093,290	B1	8/2021	Chopra et al.
2002/0035511	A1	3/2002	Haji
2002/0083079	A1	6/2002	Meier et al.
2002/0095609	A1	7/2002	Tokunaga
2002/0129047	A1	9/2002	Cane
2002/0129106	A1	9/2002	Gutfreund
2002/0194033	A1	12/2002	Huff
2002/0194511	A1	12/2002	Swoboda
2003/0140068	A1	7/2003	Yeung
2003/0200222	A1	10/2003	Feinberg
2004/0210724	A1	10/2004	Koning et al.
2005/0076251	A1	4/2005	Barr
2005/0268121	A1	12/2005	Rothman et al.
2005/0289414	A1	12/2005	Adya
2006/0058994	A1	3/2006	Ravi
2006/0101174	A1	5/2006	Kanamaru
2006/0190775	A1	8/2006	Aggarwal et al.
2006/0206507	A1	9/2006	Dahbour
2006/0224846	A1	10/2006	Amarendran
2006/0236073	A1	10/2006	Soules
2006/0242356	A1	10/2006	Mogi et al.
2006/0245411	A1	11/2006	Chen et al.
2006/0251067	A1	11/2006	Desanti
2007/0073970	A1	3/2007	Yamazaki
2007/0079156	A1	4/2007	Fujimoto
2007/0101173	A1	5/2007	Fung
2007/0168606	A1	7/2007	Takai
2007/0234302	A1	10/2007	Suzuki
2008/0005168	A1	1/2008	Huff et al.
2008/0010521	A1	1/2008	Goodrum et al.
2008/0147460	A1	6/2008	Ollivier
2008/0162592	A1	7/2008	Huang
2008/0183891	A1	7/2008	Ni
2008/0228771	A1	9/2008	Prahlad
2008/0244032	A1	10/2008	Gilson
2008/0244177	A1	10/2008	Crescenti
2008/0256384	A1	10/2008	Branson et al.
2008/0270461	A1	10/2008	Gordon
2008/0301479	A1	12/2008	Wood
2009/0077443	A1	3/2009	Nguyen et al.
2009/0198677	A1	8/2009	Sheehy
2009/0198825	A1	8/2009	Miller
2009/0210464	A1	8/2009	Chiang-Lin
2009/0268903	A1	10/2009	Bojinov et al.
2009/0282020	A1	11/2009	McSheffrey
2009/0287665	A1	11/2009	Prahlad
2009/0319534	A1	12/2009	Gokhale
2009/0327477	A1	12/2009	Madison, Jr
2010/0023722	A1	1/2010	Tabbara
2010/0064033	A1	3/2010	Travostino
2010/0070448	A1	3/2010	Omoigui
2010/0070466	A1	3/2010	Prahlad
2010/0070474	A1	3/2010	Lad
2010/0070725	A1	3/2010	Prahlad
2010/0082672	A1	4/2010	Kottomtharayil
2010/0082700	A1	4/2010	Parab
2010/0082713	A1	4/2010	Frid-Nielsen et al.
2010/0162002	A1	6/2010	Dodgson et al.
2010/0190478	A1	7/2010	Brewer
2010/0235333	A1	9/2010	Bates
2010/0257403	A1	10/2010	Virk et al.
2010/0269164	A1	10/2010	Sosnosky et al.
2010/0274772	A1	10/2010	Samuels
2010/0318782	A1	12/2010	Auradkar et al.
2010/0332401	A1	12/2010	Prahlad
2010/0333116	A1	12/2010	Prahlad
2011/0010518	A1	1/2011	Kavuri et al.
2011/0022642	A1	1/2011	DeMilo et al.
2011/0040824	A1	2/2011	Harm
2011/0055161	A1	3/2011	Wolfe
2011/0191544	A1	8/2011	Naga et al.
2011/0276713	A1	11/2011	Brand

(56) **References Cited**

U.S. PATENT DOCUMENTS

2011/0277027	A1	11/2011	Hayton et al.	
2012/0054626	A1	3/2012	Odenheimer	
2012/0084262	A1	4/2012	Dwarampudi et al.	
2012/0110186	A1	5/2012	Kapur et al.	
2012/0131645	A1	5/2012	Harm	
2012/0240183	A1	9/2012	Sinha	
2013/0007245	A1	1/2013	Malik et al.	
2013/0035795	A1	2/2013	Pfeiffer et al.	
2013/0125198	A1	5/2013	Ferguson et al.	
2013/0238572	A1	9/2013	Prahlad et al.	
2013/0238969	A1	9/2013	Smith	
2013/0262385	A1	10/2013	Vibhor et al.	
2013/0297902	A1	11/2013	Collins et al.	
2013/0326279	A1	12/2013	Chavda et al.	
2014/0189432	A1	7/2014	Gokhale et al.	
2014/0283010	A1	9/2014	Rutkowski et al.	
2014/0380014	A1	12/2014	Moyer	
2015/0113055	A1	4/2015	Vijayan et al.	
2015/0127967	A1	5/2015	Dutton	
2015/0198995	A1	7/2015	Muller et al.	
2016/0042090	A1	2/2016	Mitkar et al.	
2016/0100013	A1	4/2016	Vijayan et al.	
2016/0350391	A1	12/2016	Vijayan et al.	
2017/0039218	A1	2/2017	Prahlad	
2017/0046199	A1*	2/2017	Kramer	G06F 9/5083
2017/0126807	A1	5/2017	Vijayan et al.	
2017/0235647	A1	8/2017	Kilaru et al.	
2018/0285383	A1	10/2018	Nara	
2018/0375938	A1	12/2018	Vijayan et al.	
2019/0108341	A1	4/2019	Bedhapudi et al.	
2019/0132314	A1*	5/2019	Haravu	H04L 67/1097
2019/0171966	A1*	6/2019	Rangasamy	G06N 20/00
2019/0179805	A1	6/2019	Prahlad et al.	
2019/0182325	A1	6/2019	Vijayan et al.	
2019/0278662	A1	9/2019	Nagrale et al.	
2019/0303246	A1	10/2019	Gokhale et al.	
2020/0012569	A1*	1/2020	Natanzon	G06F 16/2255
2020/0034248	A1	1/2020	Nara	
2020/0073574	A1	3/2020	Pradhan	
2020/0117641	A1	4/2020	Mitkar et al.	
2020/0162551	A1	5/2020	Vijayan et al.	
2020/0169611	A1	5/2020	Kumar	
2020/0264775	A1	8/2020	Natanzon	
2020/0278915	A1	9/2020	Degaonkar et al.	
2020/0319694	A1	10/2020	Mohanty et al.	
2020/0349027	A1	11/2020	Bansod et al.	
2020/0351345	A1	11/2020	Bansod et al.	
2020/0394107	A1	12/2020	Rao et al.	
2021/0011808	A1	1/2021	Al-Alem et al.	
2021/0109816	A1	4/2021	Alberti et al.	

FOREIGN PATENT DOCUMENTS

EP	0467546	1/1992
EP	0774715	5/1997
EP	0809184	11/1997
EP	817040	1/1998
EP	0899662	3/1999
EP	0981090	2/2000
WO	WO9513580	5/1995
WO	WO9912098	3/1999

OTHER PUBLICATIONS

U.S. Appl. No. 16/792,026, filed Feb. 14, 2020, Kilaru et al.

U.S. Appl. No. 16/868,964, filed May 7, 2020, Mitkar et al.

U.S. Appl. No. 16/907,023, filed Jun. 19, 2020, Owen et al.

U.S. Appl. No. 16/914,020, filed Jun. 26, 2020, Amarendran et al.

U.S. Appl. No. 17/079,023, filed Oct. 23, 2020, Agrawal et al.

U.S. Appl. No. 17/114,296, filed Dec. 7, 2020, Ashraf et al.

U.S. Appl. No. 17/129,581, filed Dec. 21, 2020, Parvathamvenkatas et al.

U.S. Appl. No. 17/130,540, filed Dec. 22, 2020, Kumar et al.

U.S. Appl. No. 63/082,631, filed Sep. 24, 2020, Kavaipatti Anantharamakrishnan et al.

Armstead et al., "Implementation of a Campwide Distributed Mass Storage Service: The Dream vs. Reality," IEEE, Sep. 11-14, 1995, pp. 190-199.

Arneson, "Mass Storage Archiving in Network Environments," Digest of Papers, Ninth IEEE Symposium on Mass Storage Systems, Oct. 31, 1988-Nov. 3, 1988, pp. 45-50, Monterey, CA.

Bates, S. et al., "Sharepoint 2007 User's Guide," pp. 1-88, 2007, Springer-Verlag New York, Inc., 104 pages.

Brandon, J., "Virtualization Shakes Up Backup Strategy," <http://www.computerworld.com>, internet accessed on Mar. 6, 2008, 3 pages.

Cabrera et al., "ADSM: A Multi-Platform, Scalable, Backup and Archive Mass Storage System," Digest of Papers, Compcon '95, Proceedings of the 40th IEEE Computer Society International Conference, Mar. 5, 1995-Mar. 9, 1995, pp. 420-427, San Francisco, CA.

Chiappetta, Marco, "ESA Enthusiast System Architecture," <http://hothardware.com/Articles/NVIDIA_ESA_Enthusiast_System_Architecture/>, Nov. 5, 2007, 2 pages.

CommVault Systems, Inc., "A CommVault White Paper: VMware Consolidated Backup (VCB) Certification Information Kit," 2007, 23 pages.

CommVault Systems, Inc., "CommVault Solutions—VMware," <http://www.commvault.com/solutions/vmware/>, internet accessed Mar. 24, 2008, 2 pages.

CommVault Systems, Inc., "Enhanced Protection and Manageability of Virtual Servers," Partner Solution Brief, 2008, 6 pages.

Davis, D., "3 VMware Consolidated Backup (VCB) Utilities You Should Know," Petri IT Knowlegebase, <http://www.petri.co.il/vmware-consolidated-backup-utilities.htm>, internet accessed on Jul. 14, 2008, Jan. 7, 2008.

Davis, D., "Understanding VMware VMX Configuration Files," Petri IT Knowledgebase, <http://www.petri.co.il/virtual_vmware_vmx_configuration_files.htm>, internet accessed on Jun. 19, 2008, 6 pages.

Davis, D., "VMware Server & Workstation Disk Files Explained," Petri IT Knowledgebase, <http://www.petri.co.il/virtual_vmware_files_explained.htm>, internet accessed on Jun. 19, 2008, 5 pages.

Davis, D., "VMware Versions Compared," Petri IT Knowledgebase, <http://www.petri.co.il/virtual_vmware_versions_compared.htm>, internet accessed on Apr. 28, 2008, 6 pages.

Eitel, "Backup and Storage Management in Distributed Heterogeneous Environments," *IEEE,* Jun. 12-16, 1994, pp. 124-126.

Gait, J., "The Optical File Cabinet: A Random-Access File System for Write-Once Optical Disks," *IEEE Computer,* vol. 21, No. 6, pp. 11-22 (Jun. 1988).

International Search Report and Written Opinion for PCT/US2011/054374, dated May 2, 2012, 9 pages.

Jander, M., "Launching Storage-Area Net," *Data Communications,* US, McGraw Hill, NY, vol. 27, No. 4 (Mar. 21, 1998), pp. 64-72.

Microsoft Corporation, "How NTFS Works," Windows Server TechCenter, updated Mar. 28, 2003, internet accessed Mar. 26, 2008, 26 pages.

Rosenblum et al., "The Design and Implementation of a Log-Structured File System," *Operating Systems Review SIGOPS,* vol. 25, No. 5, New York, US, pp. 1-15 (May 1991).

Sanbarrow.com, "Disktype-table," <http://sanbarrow.com/vmdk/disktypes.html>, internet accessed on Jul. 22, 2008, 4 pages.

Sanbarrow.com, "Files Used by a VM," <http://sanbarrow.com/vmx/vmx-files-used-by-a-vm.html>, internet accessed on Jul. 22, 2008, 2 pages.

Sanbarrow.com, "Monolithic Versus Split Disks," <http://sanbarrow.com/vmdk/monolithicversusspllit.html>, internet accessed on Jul. 14, 2008, 2 pages.

VMware, Inc., "Open Virtual Machine Format," <http://www.vmware.com/appliances/learn/ovf.html>, internet accessed on May 6, 2008, 2 pages.

VMware, Inc., "OVF, Open Virtual Machine Format Specification, version 0.9," White Paper, <http://www.vmware.com>, 2007, 50 pages.

(56) **References Cited**

OTHER PUBLICATIONS

VMware, Inc., "The Open Virtual Machine Format Whitepaper for OVF Specification, version 0.9," White Paper, <http://www.vmware.com>, 2007, 16 pages.
VMware, Inc., "Understanding VMware Consolidated Backup," White Paper, <http://www.vmware.com>, 2007, 11 pages.
VMware, Inc., "Using VMware Infrastructure for Backup and Restore," Best Practices, <http://www.vmware.com>, 2006, 20 pages.
VMware, Inc., "Virtual Disk API Programming Guide," <http://www.vmware.com>, Revision 20080411, 2008, 44 pages.
VMware, Inc., "Virtual Disk Format 1.1," VMware Technical Note, <http://www.vmware.com>, Revision 20071113, Version 1.1, 2007, 18 pages.
VMware, Inc., "Virtual Machine Backup Guide, ESX Server 3.0.1 and VirtualCenter 2.0.1," <http://www.vmware.com>, updated Nov. 21, 2007, 74 pages.
VMware, Inc., "Virtual Machine Backup Guide, ESX Server 3.5, ESX Server 3i version 3.5, VirtualCenter 2.5," <http://www.vmware.com>, updated Feb. 21, 2008, 78 pages.
VMware, Inc., "Virtualized iSCSI SANS: Flexible, Scalable Enterprise Storage for Virtual Infrastructures," White Paper, <http://www.vmware.com>, Mar. 2008, 13 pages.
VMware, Inc., "VMware Consolidated Backup, Improvements in Version 3.5," Information Guide, <http://www.vmware.com>, 2007, 11 pages.
VMware, Inc., "VMware Consolidated Backup," Product Datasheet, <http://www.vmware.com>, 2007, 2 pages.
VMware, Inc., "VMware ESX 3.5," Product Datasheet, <http://www.vmware.com>, 2008, 4 pages.
VMware, Inc., "VMware GSX Server 3.2, Disk Types: Virtual and Physical," <http://www.vmware.com/support/gsx3/doc/disks_types_gsx.html>, internet accessed on Mar. 25, 2008, 2 pages.
VMware, Inc., "VMware OVF Tool," Technical Note, <http://www.vmware.com>, 2007, 4 pages.
VMware, Inc., "VMware Workstation 5.0, Snapshots in a Linear Process," <http:/www.vmware.com/support/ws5/doc/ws_preserve_sshot_linear.html>, internet accessed on Mar. 25, 2008, 1 page.
VMware, Inc., "VMware Workstation 5.0, Snapshots in a Process Tree," <http://www.vmware.com/support/ws5/doc/ws_preserve_sshot_tree.html>, internet accessed on Mar. 25, 2008, 1 page.
VMware, Inc., "VMware Workstation 5.5, What Files Make Up a Virtual Machine?" <http://www.vmware.com/support/ws55/doc/ws_learning_files_in_a_vm.html>, internet accessed on Mar. 25, 2008, 2 pages.
Wikipedia, "Cloud computing," <http://en.wikipedia.org/wiki/Cloud_computing>, internet accessed Jul. 8, 2009, 13 pages.
Wikipedia, "Cluster (file system)," <http://en.wikipedia.org/wiki/Cluster_%28file_system%29>, internet accessed Jul. 25, 2008, 1 page.
Wikipedia, "Cylinder-head-sector," <http://en.wikipedia.org/wiki/Cylinder-head-sector>, internet accessed Jul. 22, 2008, 6 pages.
Wikipedia, "File Allocation Table," <http://en.wikipedia.org/wiki/File_Allocation_Table>, internet accessed on Jul. 25, 2008, 19 pages.
Wikipedia, "Logical Disk Manager," <http://en.wikipedia.org/wiki/Logical_Disk_Manager>, internet accessed Mar. 26, 2008, 3 pages.
Wikipedia, "Logical Volume Management," <http://en.wikipedia.org/wiki/Logical_volume_management>, internet accessed on Mar. 26, 2008, 5 pages.
Wikipedia, "Storage Area Network," <http://en.wikipedia.org/wiki/Storage_area_network>, internet accessed on Oct. 24, 2008, 5 pages.
Wikipedia, "Virtualization," <http://en.wikipedia.org/wiki/Virtualization>, internet accessed Mar. 18, 2008, 7 pages.
U.S. Appl. No. 16/709,579, Holistically Protecting Serverless Applications Distributed Across One or More Cloud Computing Environments, filed Dec. 10, 2019.
Case No. No. 6:21-CV-00634-ADA, Answer WDTX-6-21-CV-00634-19, filed Aug. 27, 2021, in 23 pages.
Case No. 1:21-cv-00537, Complaint WDTX-1-21-cv-00537-1_WDTX-6-21-cv-00634-1, filed Jun. 18, 2021, in 44 pages.
Case No. 6:21-cv-00634-ADA, Order Dismissing with Prejudice WDTX-6-21-CV-00634-22, filed Sep. 1, 2021, in 1 page.
Case No. 1:20-cv-00525-MN, First Amended Answer DDE-1-20-cv-00525-95, filed Jul. 23, 2021, in 38 pages.
Case No. 1-20-cv-00525, Oral Order DDE-1-20-cv-00524-86_DDE-1-20-cv-00525-87, filed Jun. 29, 2021, in 1 page.
Case No. 1-20-CV-00525-MN, Oral Order DDE-1-20-cv-00524-78_DDE-1-20-cv-00525-77, filed May 24, 2021, in 1 page.
Case No. 1:20-cv-00525-MN, Order DDE-1-20-cv-00525-38_DDE-1-20-cv-00524-42, filed Feb. 10, 2021, in 4 pages.
Case No. 1:20-cv-00525-MN, Amended Complaint DDE-1-20-CV-00525-15, filed Jul. 27, 2020, in 30 pages.
Case No. 1 :20-cv-00525-MN, Complaint DDE-1-20-cv-00525-1, filed Apr. 21, 2020, in 28 pages.
Case No. C.A. No. 20-525-MN-CJB, Joint Claim Construction Brief Exhibits DDE-1 -20-cv-00525-107-1, filed Oct. 1, 2021, in 488 pages (in 7 parts).
Case No. 1:20-cv-00525-MN-CJB, Oral Order DDE-1-20-cv-00524-78_DDE-1-20-cv-00525-77, dated May 24, 2021, in 1 page.
Case No. C.A. No. 20-525-MN-CJB, Joint Claim Construction Brief DDE-1-20-cv-00525-107, filed October 1, in 79 pages.
Case No. C.A. No. 20-525-MN-CJB, Joint Claim Construction Statement DDE-1-20-cv-00525-119, filed Oct. 29, 2021, in 12 pages.
Case No. C.A. No. 20-525-MN-CJB, Farnan Letter DDE-1-20-cv-00525-111, filed Oct. 6, 2021, in 2 pages.
Case No. C.A. No. 20-525-MN-CJB, Farnan Letter Exibit A DDE-1-20-CV-00525-111-1, filed Oct. 6, 2021, in 7 pages.
Case No. 1:20-cv-00525-CFC-CJB, Joint Appendix of Exhibits 1-6, filed Jan. 13, 2022, in 2 pages.
Case No. 1:20-cv-00525-CFC-CJB, Joint Appendix of Exhibits 1-6, filed Jan. 13, 2022, in 224 pages.
Case No. 1:20-cv-00525-CFC-CJB, Joint Claim Construction Brief On Remaining Disputed Terms, filed Jan. 13, 2022, in 54 pages.
Case No. 120-CV-00525-MN—Stipulation of Dismissal, filed Jan. 27, 2022, in 2 pages.
Case No. 1:20-cv-00524-MN, Order Dismissing with Prejudice DDE-1-20-cv-00524-101, filed Aug. 31, 2021, in 1 page.
Case No. 1:20-cv-00524-MN, First Amended Answer DDE-1-20-cv-00524-96, filed Jul. 23, 2021, in 41 pages.
Case No. 1:20-cv-00524-MN, Stipulation DDE-1-20-cv-00524-93, filed Jul. 14, 2021, in 3 pages.
Case No. 1:20-cv-00524-MN, Oral Order DDE-1-20-cv-00524-86_DDE-1-20-cv-00525-87, filed Jun. 29, 2021, in 1 page.
Case No. 1:20-cv-00524-MN, Answer DDE-1-20-cv-00524-45, filed Feb. 16, 2021, in 25 pages.
Case No. 1:20-cv-00524-MN, Order DDE-1-20-cv-00525-38_DDE-1-20-cv-00524-42, filed Feb. 10, 2021, in 4 pages.
Case No. 1:20-cv-00524-MN, Amended Complaint DDE-1-20-CV-00524-13, filed Jul. 27, 2020, in 30 pages.
Case No. 1 :20-cv-00524-MN, Complaint DDE-1-20-cv-00524-1, filed Apr. 21, 2020, in 29 pages.
PTAB-IPR2021-00675—00589 590 675 Termination Order, filed Sep. 1, 2021, in 4 pages.
PTAB-IPR2021-00675—Joint Motion to Terminate, Aug. 31, 2021, in 7 pages.
PTAB-IPR2021-00675—Joint Request to Seal Settlement Agreement, Aug. 31, 2021, in 4 pages.
PTAB-IPR2021-00675—Preliminary Sur-Reply Final, filed Aug. 16, 2021, in 6 pages.
PTAB-IPR2021-00675—Reply to POPR, filed Aug. 9, 2021, in 6 pages.
PTAB-IPR2021-00675—POPR Final, filed Jul. 9, 2021, in 48 pages.
PTAB-IPR2021-00675—Mar. 25, 2021 IPR—Petition—Cls 5 and 21—Final, dated Mar. 25, 2021, in 72 pages.
PTAB-IPR2021-00675—Exhibit 1001—U.S. Pat. No. 10,248,657, issue date Apr. 2, 2019, in 85 pages.
PTAB-IPR2021-00675—Exhibit 1002—Jagadish_Declaration_Final, filed Mar. 24, 2021, in 175 pages.

(56) **References Cited**

OTHER PUBLICATIONS

PTAB-IPR2021-00675—Exhibit 1003—WO2008070688A1 (Bunte), dated Jun. 12, 2008, in 71 pages.
PTAB-IPR2021-00675—Exhibit 1004—US20070156842A1 (Vermeulen), Publication Date Jul. 5, 2007, 1 in 69 pages.
PTAB-IPR2021-00675—Exhibit 1005—US20020059317A1 (Black), Publication Date May 16, 2002, in 14 pages.
PTAB-IPR2021-00675—Exhibit 1006—US_20080133835 (Zhu), Publication Date Jun. 5, 2008, in 14 pages.
PTAB-IPR2021-00675—Exhibit 1007—Introduction to AWS for Java Developers (Monson-Haefel), Jun. 26, 2007, in 3 pages.
PTAB-IPR2021-00675—Exhibit 1008—U.S. Pat. No. 8,140,786 (Bunte Patent), Issue Date Mar. 20, 2012, in 37 pages.
PTAB-IPR2021-00675—Exhibit 1009—US20060218435A1 (Ingen), Publication Date Sep. 28, 2006, in 27 pages.
PTAB-IPR2021-00675—Exhibit 1010—Declaration of Duncan Hall, dated Feb. 18, 2021, in 81 pages.
PTAB-IPR2021-00675—Exhibit 1011—Controlling the Enterprise Information Life Cycle, Jun. 10, 2005, in 7 pages.
PTAB-IPR2021-00675—Exhibit 1012—Deduplication_Stop repeating yourself_Network World, Sep. 25, 2006, in 6 pages.
PTAB-IPR2021-00675—Exhibit 1013—Lose Unwanted Gigabytes Overnight (McAdams), Feb. 26, 2007, in 5 pages.
PTAB-IPR2021-00675—Exhibit 1014—Data Domain releases DD120 for backup and deduplication_Network World, Feb. 26, 2008, in 2 pages.
PTAB-IPR2021-00675—Exhibit 1015—PTAB-IPR2021-00675—Exhibit 1015—Hall-Ellis Declaration_Part1, dated Feb. 24, 2021, in 299 pages of Part 1 of 5.
PTAB-IPR2021-00675—Exhibit 1015—Hall-Ellis Declaration_Part2, dated Feb. 24, 2021, in 306 pages of Part 2 of 5.
PTAB-IPR2021-00675—Exhibit 1015—Hall-Ellis Declaration_Part3, dated Feb. 24, 2021, in 272 pages of Part 3 of 5.
PTAB-IPR2021-00675—Exhibit 1015—Hall-Ellis Declaration_Part4, dated Feb. 24, 2021, in 364 pages of Part 4 of 5.
PTAB-IPR2021-00675—Exhibit 1015—Hall-Ellis Declaration_Part5., dated Feb. 24, 2021, in 480 pages of Part 5 of 5.
PTAB-IPR2021-00675—Exhibit 1016—Amazon.com Unveils Data Storage Service_Computerworld, Mar. 20, 2006, in 4 pages.
PTAB-IPR2021-00675—Exhibit 1017—Amazon Simple Storage Service (Amazon S3)b, Mar. 10, 2008, in 2 pages.
PTAB-IPR2021-00675—Exhibit 1018—How I cut my data center costs by $700,000_Computerworld, Mar. 30, 2007, in 4 pages.
PTAB-IPR2021-00675—Exhibit 1019—Microsoft Office Outlook 2003 (Boyce)_Part1,2004, in 269 pages, Part 1 of 4.
PTAB-IPR2021-00675—Exhibit 1019—Microsoft Office Outlook 2003 (Boyce)_Part2, 2004, in 303 pages, Part 2 of 4.
PTAB-IPR2021-00675—Exhibit 1019—Microsoft Office Outlook 2003 (Boyce)_Part3, 2004, in 274 pages, Part 3 of 4.
PTAB-IPR2021-00675—Exhibit 1019—Microsoft Office Outlook 2003 (Boyce) Part4, in 220 pages, Part 4 of 4.
PTAB-IPR2021-00675—Exhibit 1020—Finding Similar Files in a Large File System (Udi), Oct. 1993 in 11 pages.
PTAB-IPR2021-00675—Exhibit 1021—Components of Amazon S3, Mar. 1, 2006, in 1 page.
PTAB-IPR2021-00675—Exhibit 1022—Apr. 21, 2020 [1] Complaint, filed Apr. 21, 2020, in 300 pages.
PTAB-IPR2021-00675—Exhibit 1023—Feb. 17, 2021 Scheduling Order Case, filed Feb. 17, 2021, in 15 pages.
PTAB-IPR2021-00675—Exhibit 1024—Corporate IT Warms Up to Online Backup Services_Computerworld, Feb. 4, 2008, in 6 pages.
PTAB-IPR2021-00675—Exhibit 1025—Who Are The Biggest Users of Amazon Web Services Its Not Startups TechCrunch 1, Apr. 21, 2008, in 2 pages.
PTAB-IPR2021-00675—Exhibit 1026—Programming Amazon Web Services (Murty), Mar. 2008, OReilly Media, Inc. IN 595 pages.
PTAB-IPR2021-00675—Exhibit 1027—Announcement Lower Data Transfer Costs, posted on Apr. 22, 2008, in 1 page.
PTAB-IPR2021-00675—Exhibit 1028—A Ruby Library for Amazons Simple Storage Services (S3), May 26, 2008, 8 pages.
PTAB-IPR2021-00675—Exhibit 1029—Erlaws. Mar. 7, 2008, in 2 pages.
PTAB-IPR2021-00675—Exhibit 1030—S3Drive—Amazon S3 Filesystem, Jun. 16, 2008, in 1 page.
PTAB-IPR2021-00675—Exhibit 1031—S3Drive—Prerequisites, Jun. 5, 2008, in 1 page.
PTAB-IPR2021-00675—Exhibit 1032—S3Drive Screenshots, Jun. 21, 2008, in 1 page.
PTAB-IPR2021-00675—Exhibit 1033—S3Drive Download, Jun. 5, 2008, 1 page.
PTAB-IPR2021-00675—Exhibit 1034—ElasticDrive, Jun. 11, 2008, in 1 page.
PTAB-IPR2021-00675—Exhibit 1035—Jungle Disk Overview, Jun. 26, 2008, in 1 page.
PTAB-IPR2021-00675—Exhibit 1036—Jungle Disk How It Works, Jun. 26, 2008, in 1 page.
PTAB-IPR2021-00675—Exhibit 1037—Jungle Disk Why Its Better, Jun. 26, 2008, in 1 page.
PTAB-IPR2021-00675—Exhibit 1038—Jungle Disk FAQs, Jun. 26, 2008, in 1 page.
PTAB-IPR2021-00675—Exhibit 1039—IBiz Amazon Integrator, May 17, 2008, in 3 pages.
PTAB-IPR2021-00675—Exhibit 1040—JetS3t, May 29, 2008, in 1 page.
PTAB-IPR2021-00675—Exhibit 1041—SpaceBlock, Jun. 4, 2008, in 2 pages.
PTAB-IPR2021-00675—Exhibit 1042—S3Safe, May 31, 2008, 1 page.
PTAB-IPR2021-00675—Exhibit 1043—Veritas NetBackup, Jun. 22, 2008, in 1 page.
PTAB-IPR2021-00675—Exhibit 1044—Amazon S3 Application Programming Interfaces, Mar. 1, 2006, in 1 page.
PTAB-IPR2021-00675—Exhibit 1045—Bucket Restrictions and Limitations, Mar. 1, 2006, in 1 page.
PTAB-IPR2021-00675—Exhibit 1046—The Archive in the Sky, posted on Jun. 17, 2008, in 2 pages.
PTAB-IPR2021-00675—Exhibit 1047—Market-Oriented Cloud Computing (Buyya), Sep. 25-27, 2008, in 9 pages.
PTAB-IPR2021-00675—Exhibit 1048—Data Domain OpenStorage Software, May 12, 2008, 1 page.
PTAB-IPR2021-00675—Exhibit 1049—Working With Amazon Buckets, in 1 page.
PTAB-IPR2021-00675—Exhibit 1050—FH 10248657, Issue Date Apr. 2, 2019, in 677 pages.
PTAB-IPR2021-00675—Exhibit 1051—*CommVault* v. *Cohesity* Complaint, dated Apr. 21, 2020, in 28 pages.
PTAB-IPR2021-00675—Exhibit 1052—US_2008_0052328_A1 (Widhelm), Publication Date Feb. 28, 2008, in 10 pages.
PTAB-IPR2021-00675—Exhibit 1053—Microsoft Computer Dictionary, 2002, in 12 pages.
PTAB-IPR2021-00675—Exhibit 2001—US20080229037A1, Publication Date Sep. 18, 2008, in 36 pages.
PTAB-IPR2021-00675—Exhibit 2002—Comparison of Exs. 1008, 1003, in 75 pages.
PTAB-IPR2021-00675—Exhibit 3001—RE_IPR2021-00535, 2021-00589, 2021-00590, 2021-00609, 2021-00673, 2021-00674, 2021-00675, Aug. 30, 2021, in 2 pages.

* cited by examiner

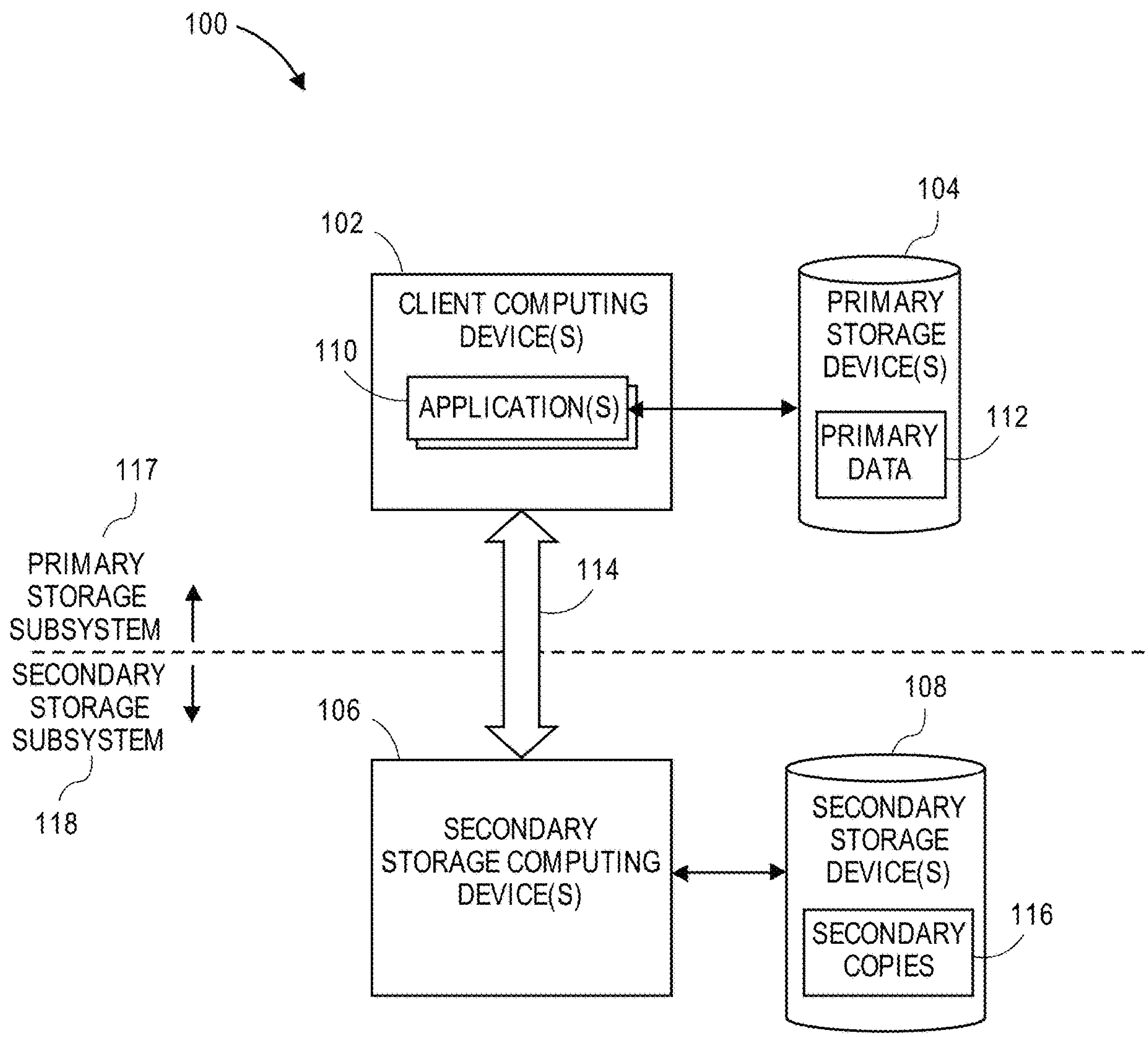
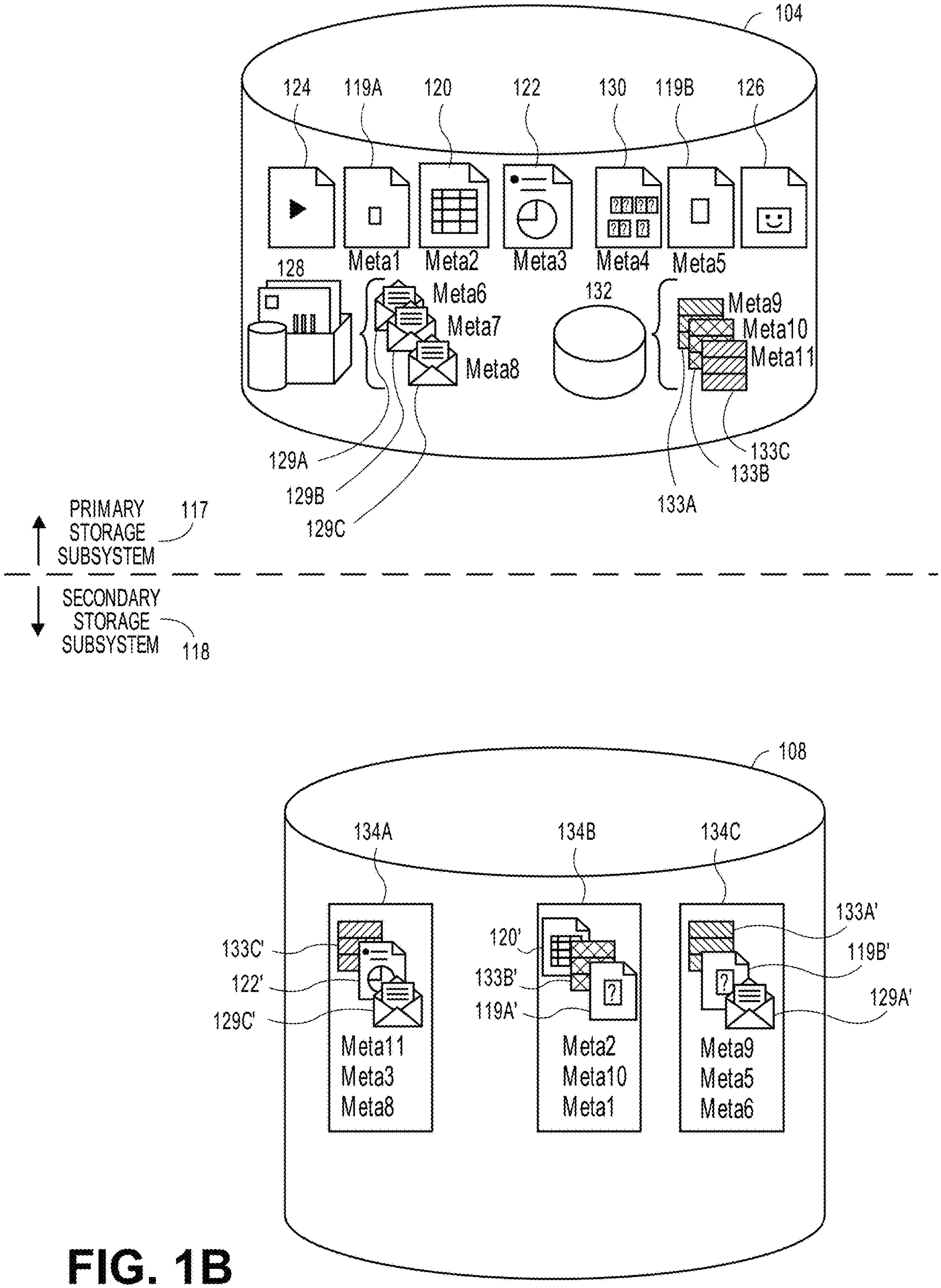

100
102
CLIENT COMPUTING DEVICE(S)
110
APPLICATION(S)
104
PRIMARY STORAGE DEVICE(S)
PRIMARY DATA
112
117
PRIMARY STORAGE SUBSYSTEM
114
SECONDARY STORAGE SUBSYSTEM
118
106
SECONDARY STORAGE COMPUTING DEVICE(S)
108
SECONDARY STORAGE DEVICE(S)
SECONDARY COPIES
116

FIG. 1A



104
124
119A
120
122
130
119B
126
Meta1
Meta2
Meta3
Meta4
Meta5
128
Meta6
Meta7
Meta8
132
Meta9
Meta10
Meta11
129A
129B
129C
133C
133B
133A
PRIMARY STORAGE SUBSYSTEM
117
SECONDARY STORAGE SUBSYSTEM
118
108
134A
134B
134C
133C'
122'
129C'
Meta11
Meta3
Meta8
120'
133B'
119A'
Meta2
Meta10
Meta1
133A'
119B'
129A'
Meta9
Meta5
Meta6

FIG. 1B

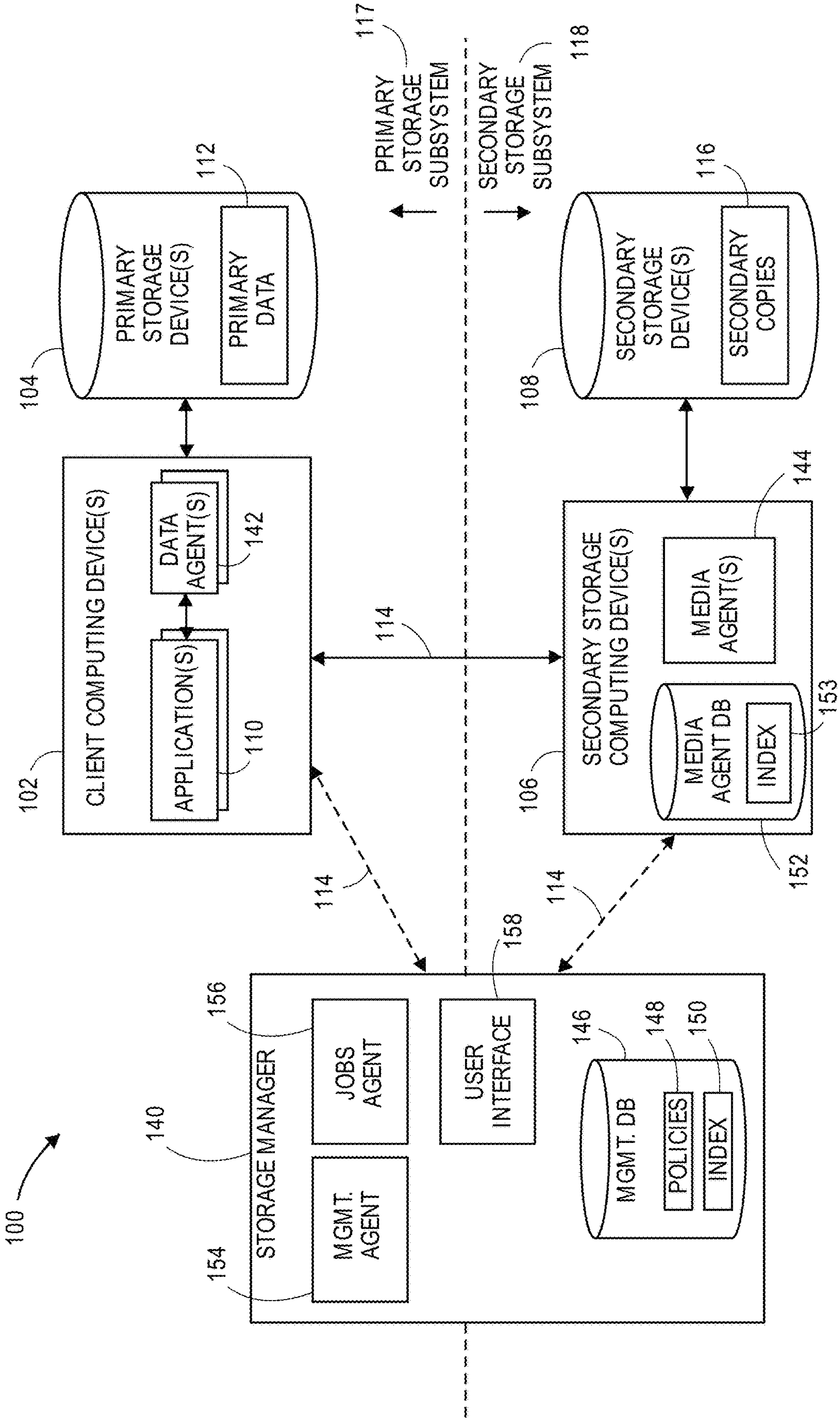

100
CLIENT COMPUTING DEVICE(S)
102
APPLICATION(S)
110
DATA AGENT(S)
142
PRIMARY STORAGE DEVICE(S)
104
PRIMARY DATA
112
PRIMARY STORAGE SUBSYSTEM
117
SECONDARY STORAGE SUBSYSTEM
118
114
SECONDARY STORAGE COMPUTING DEVICE(S)
106
MEDIA AGENT(S)
144
MEDIA AGENT DB
152
INDEX
153
SECONDARY STORAGE DEVICE(S)
108
SECONDARY COPIES
116
STORAGE MANAGER
140
MGMT. AGENT
154
JOBS AGENT
156
USER INTERFACE
158
MGMT. DB
146
POLICIES
148
INDEX
150

FIG. 1C

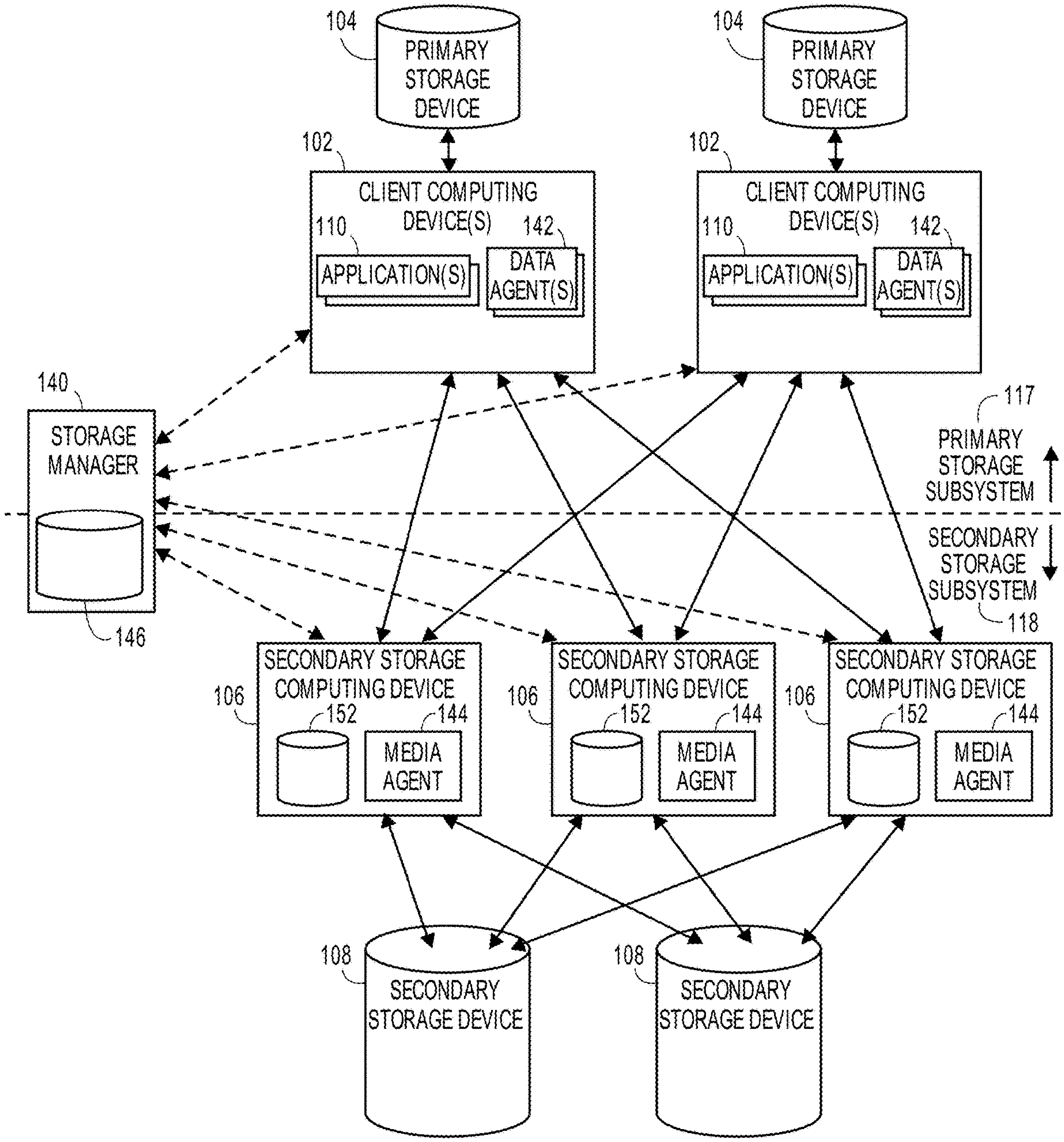
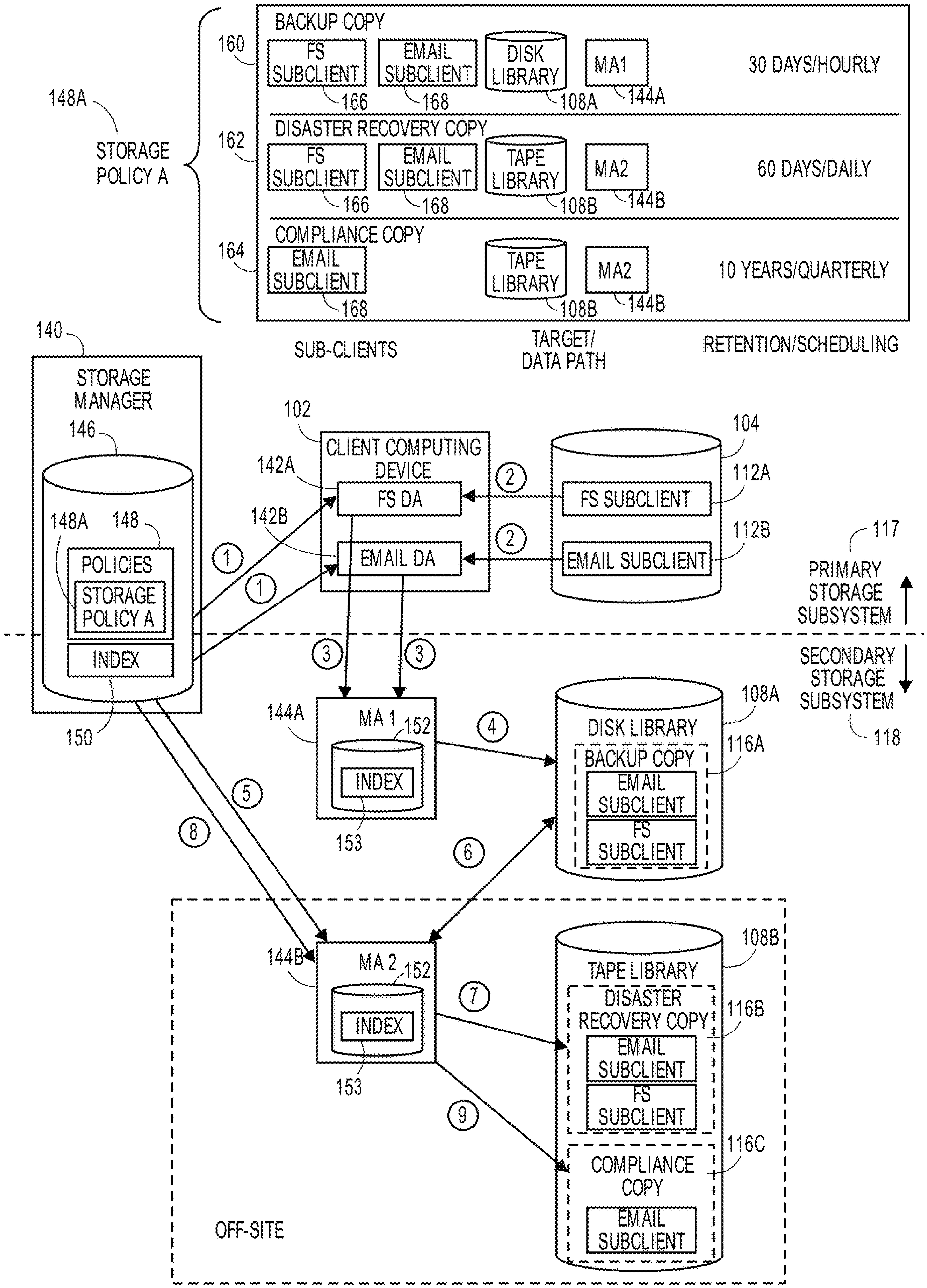

104
PRIMARY STORAGE DEVICE
104
PRIMARY STORAGE DEVICE
102
CLIENT COMPUTING DEVICE(S)
110
APPLICATION(S)
142
DATA AGENT(S)
102
CLIENT COMPUTING DEVICE(S)
110
APPLICATION(S)
142
DATA AGENT(S)
140
STORAGE MANAGER
146
117
PRIMARY STORAGE SUBSYSTEM
SECONDARY STORAGE SUBSYSTEM
118
106
SECONDARY STORAGE COMPUTING DEVICE
152
144
MEDIA AGENT
106
SECONDARY STORAGE COMPUTING DEVICE
152
144
MEDIA AGENT
106
SECONDARY STORAGE COMPUTING DEVICE
152
144
MEDIA AGENT
108
SECONDARY STORAGE DEVICE
108
SECONDARY STORAGE DEVICE

FIG. 1D



148A
STORAGE POLICY A
160
BACKUP COPY
FS SUBCLIENT
166
EMAIL SUBCLIENT
168
DISK LIBRARY
108A
MA1
144A
30 DAYS/HOURLY
162
DISASTER RECOVERY COPY
FS SUBCLIENT
166
EMAIL SUBCLIENT
168
TAPE LIBRARY
108B
MA2
144B
60 DAYS/DAILY
164
COMPLIANCE COPY
EMAIL SUBCLIENT
168
TAPE LIBRARY
108B
MA2
144B
10 YEARS/QUARTERLY
SUB-CLIENTS
TARGET/ DATA PATH
RETENTION/SCHEDULING
140
STORAGE MANAGER
146
148A
148
POLICIES
STORAGE POLICY A
INDEX
150
102
CLIENT COMPUTING DEVICE
142A
FS DA
142B
EMAIL DA
104
112A
FS SUBCLIENT
112B
EMAIL SUBCLIENT
117
PRIMARY STORAGE SUBSYSTEM
SECONDARY STORAGE SUBSYSTEM
118
1
2
3
4
5
6
7
8
9
144A
MA 1
152
INDEX
153
108A
DISK LIBRARY
116A
BACKUP COPY
EMAIL SUBCLIENT
FS SUBCLIENT
144B
MA 2
152
INDEX
153
108B
TAPE LIBRARY
116B
DISASTER RECOVERY COPY
EMAIL SUBCLIENT
FS SUBCLIENT
116C
COMPLIANCE COPY
EMAIL SUBCLIENT
OFF-SITE

FIG. 1E

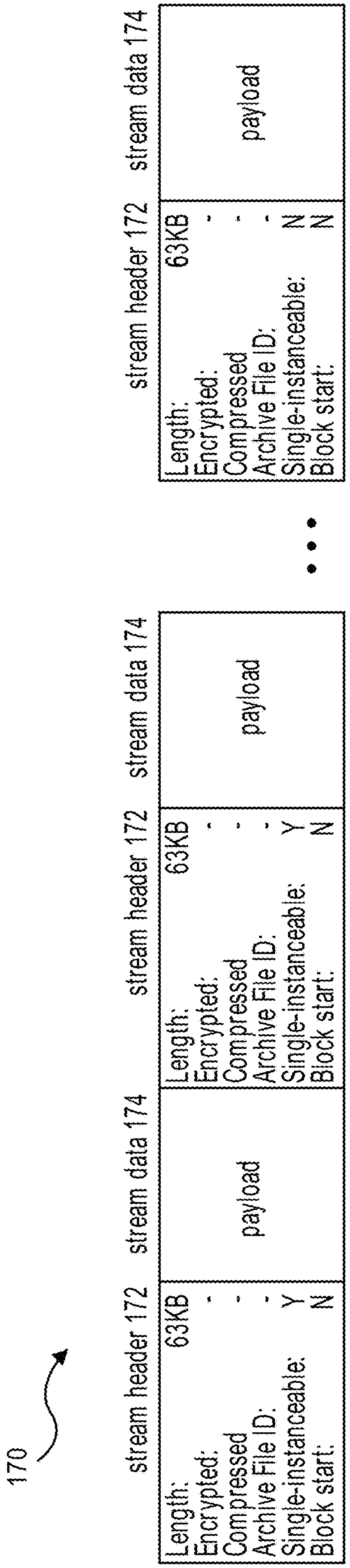

170
stream header 172
stream data 174
Length: 63KB
Encrypted: -
Compressed -
Archive File ID: -
Single-instanceable: Y
Block start: N
payload
Single-instanceable: N

FIG. 1F

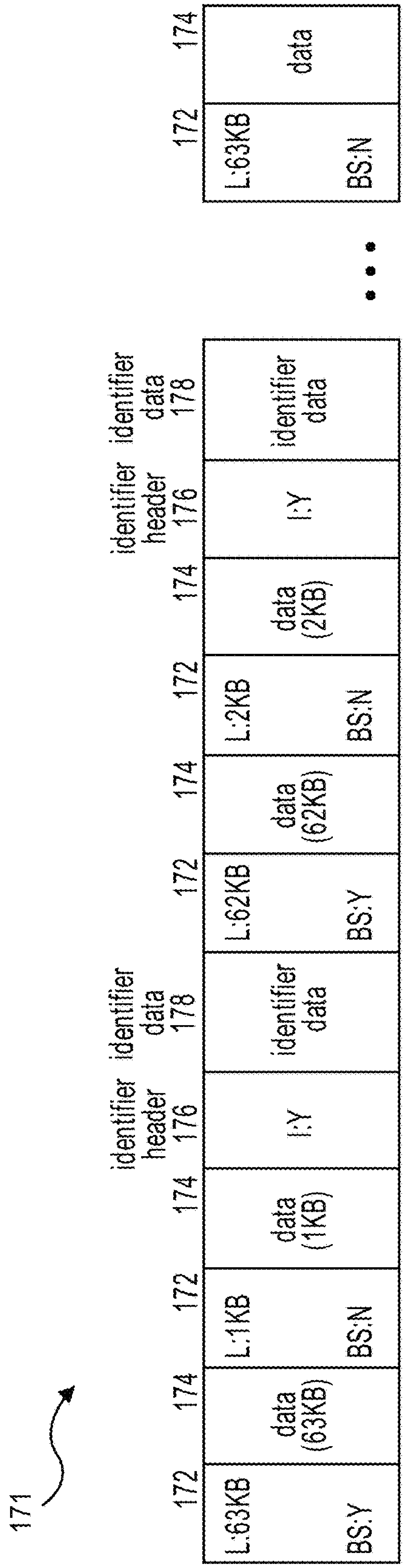

171
172
174
L:63KB
BS:Y
data (63KB)
L:1KB
BS:N
data (1KB)
identifier header 176
I:Y
identifier data 178
identifier data
L:62KB
data (62KB)
L:2KB
data (2KB)
data

FIG. 1G

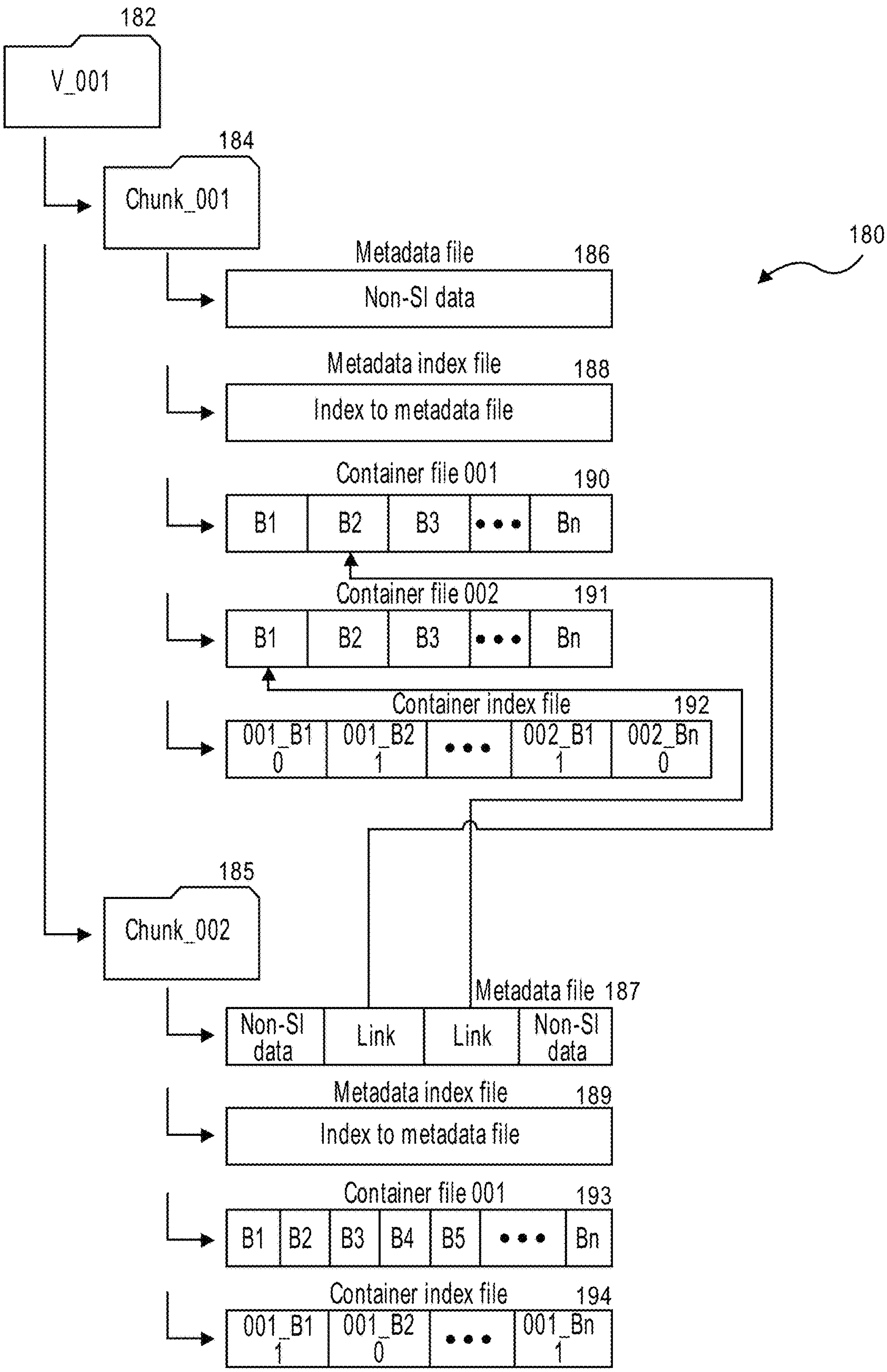

182
V_001
184
Chunk_001
180
Metadata file 186
Non-SI data
Metadata index file 188
Index to metadata file
Container file 001 190
B1
B2
B3
Bn
Container file 002 191
B1
B2
B3
Bn
Container index file 192
001_B1 0
001_B2 1
002_B1 1
002_Bn 0
185
Chunk_002
Metadata file 187
Non-SI data
Link
Link
Non-SI data
Metadata index file 189
Index to metadata file
Container file 001 193
B1
B2
B3
B4
B5
Bn
Container index file 194
001_B1 1
001_B2 0
001_Bn 1

FIG. 1H

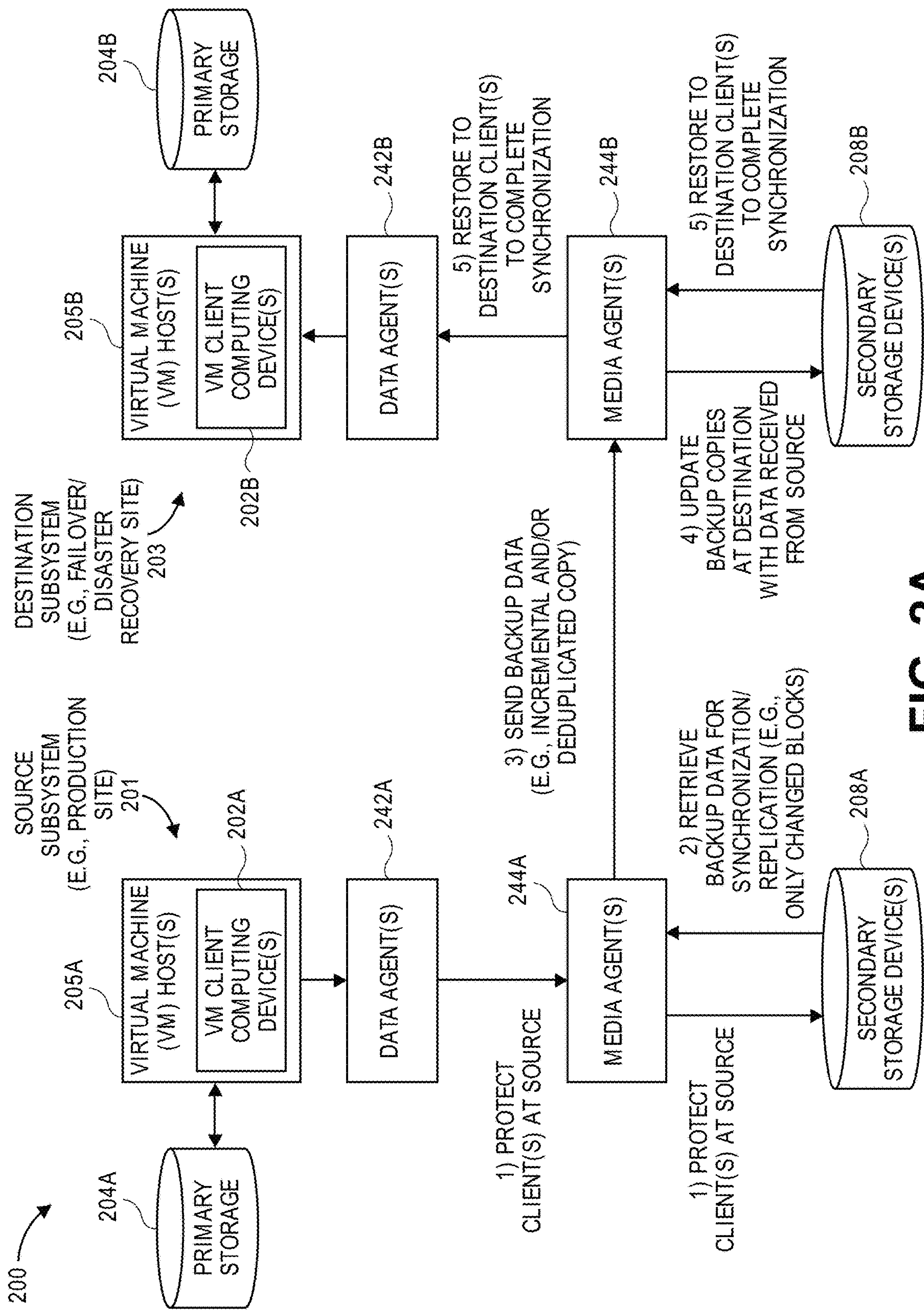

200
SOURCE SUBSYSTEM (E.G., PRODUCTION SITE) 201
DESTINATION SUBSYSTEM (E.G., FAILOVER/ DISASTER RECOVERY SITE) 203
204A
PRIMARY STORAGE
205A
VIRTUAL MACHINE (VM) HOST(S)
202A
VM CLIENT COMPUTING DEVICE(S)
242A
DATA AGENT(S)
244A
MEDIA AGENT(S)
208A
SECONDARY STORAGE DEVICE(S)
1) PROTECT CLIENT(S) AT SOURCE
1) PROTECT CLIENT(S) AT SOURCE
2) RETRIEVE BACKUP DATA FOR SYNCHRONIZATION/ REPLICATION (E.G., ONLY CHANGED BLOCKS)
3) SEND BACKUP DATA (E.G., INCREMENTAL AND/OR DEDUPLICATED COPY)
4) UPDATE BACKUP COPIES AT DESTINATION WITH DATA RECEIVED FROM SOURCE
5) RESTORE TO DESTINATION CLIENT(S) TO COMPLETE SYNCHRONIZATION
5) RESTORE TO DESTINATION CLIENT(S) TO COMPLETE SYNCHRONIZATION
204B
PRIMARY STORAGE
205B
VIRTUAL MACHINE (VM) HOST(S)
202B
VM CLIENT COMPUTING DEVICE(S)
242B
DATA AGENT(S)
244B
MEDIA AGENT(S)
208B
SECONDARY STORAGE DEVICE(S)

FIG. 2A

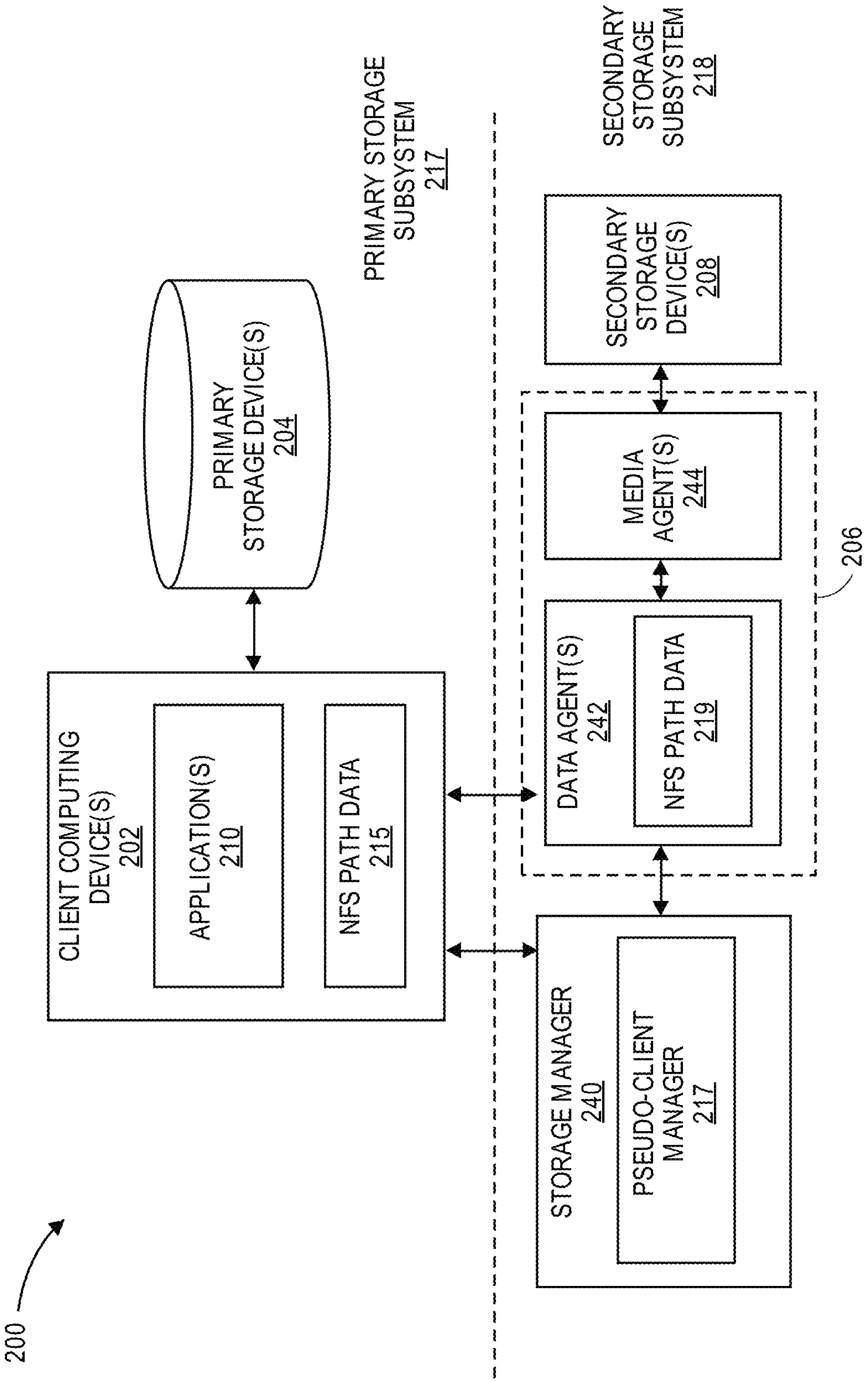

200
PRIMARY STORAGE SUBSYSTEM 217
SECONDARY STORAGE SUBSYSTEM 218
PRIMARY STORAGE DEVICE(S) 204
CLIENT COMPUTING DEVICE(S) 202
APPLICATION(S) 210
NFS PATH DATA 215
SECONDARY STORAGE DEVICE(S) 208
MEDIA AGENT(S) 244
DATA AGENT(S) 242
NFS PATH DATA 219
206
STORAGE MANAGER 240
PSEUDO-CLIENT MANAGER 217

FIG. 2B

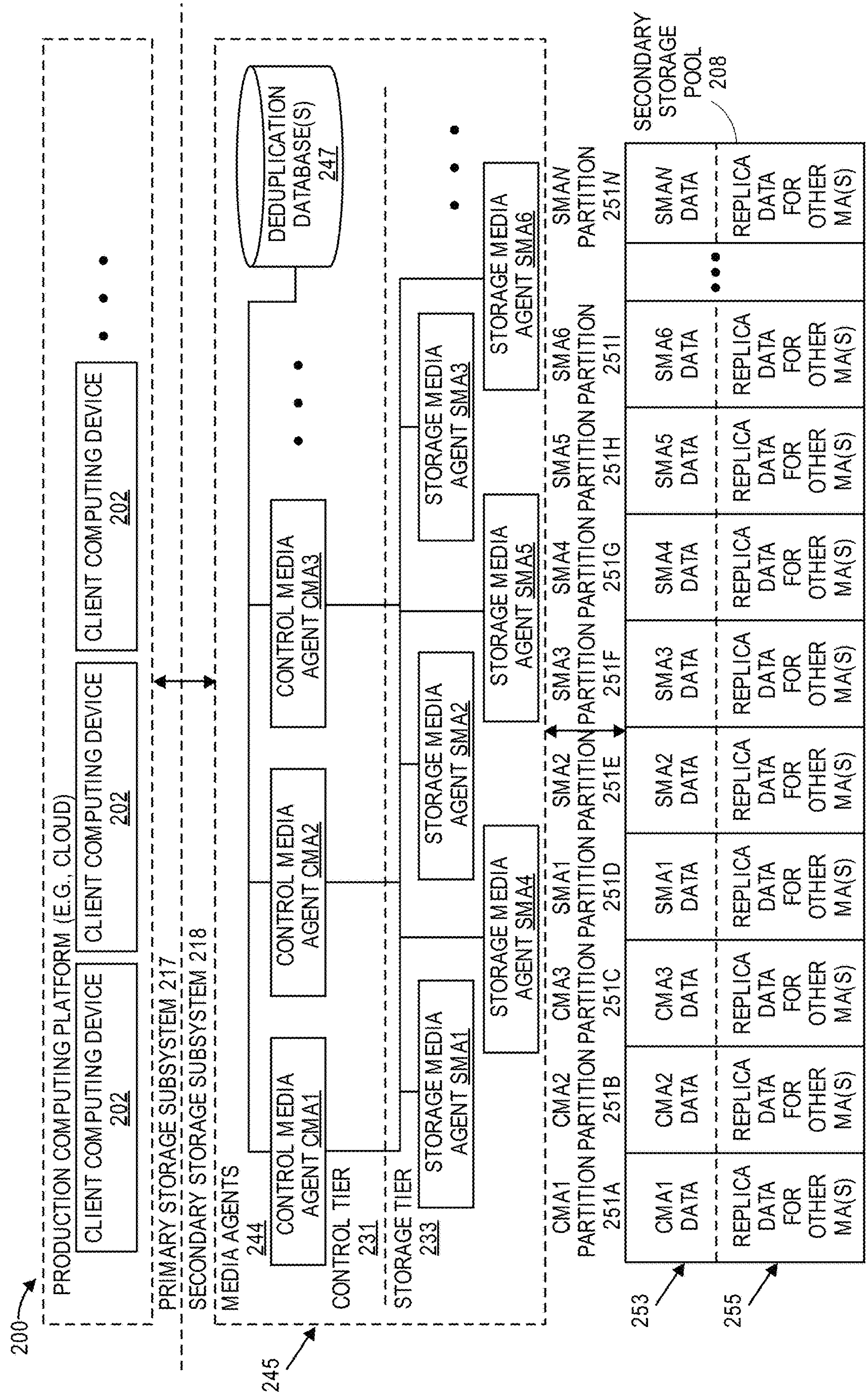

200
PRODUCTION COMPUTING PLATFORM (E.G., CLOUD)
CLIENT COMPUTING DEVICE 202
CLIENT COMPUTING DEVICE 202
CLIENT COMPUTING DEVICE 202
PRIMARY STORAGE SUBSYSTEM 217
SECONDARY STORAGE SUBSYSTEM 218
245
MEDIA AGENTS 244
DEDUPLICATION DATABASE(S) 247
CONTROL MEDIA AGENT CMA1
CONTROL MEDIA AGENT CMA2
CONTROL MEDIA AGENT CMA3
CONTROL TIER 231
STORAGE TIER 233
STORAGE MEDIA AGENT SMA1
STORAGE MEDIA AGENT SMA2
STORAGE MEDIA AGENT SMA3
STORAGE MEDIA AGENT SMA4
STORAGE MEDIA AGENT SMA5
STORAGE MEDIA AGENT SMA6
CMA1 PARTITION 251A
CMA2 PARTITION 251B
CMA3 PARTITION 251C
SMA1 PARTITION 251D
SMA2 PARTITION 251E
SMA3 PARTITION 251F
SMA4 PARTITION 251G
SMA5 PARTITION 251H
SMA6 PARTITION 251I
SMAN PARTITION 251N
CMA1 DATA
CMA2 DATA
CMA3 DATA
SMA1 DATA
SMA2 DATA
SMA3 DATA
SMA4 DATA
SMA5 DATA
SMA6 DATA
SMAN DATA
REPLICA DATA FOR OTHER MA(S)
253
255
SECONDARY STORAGE POOL 208

FIG. 2C

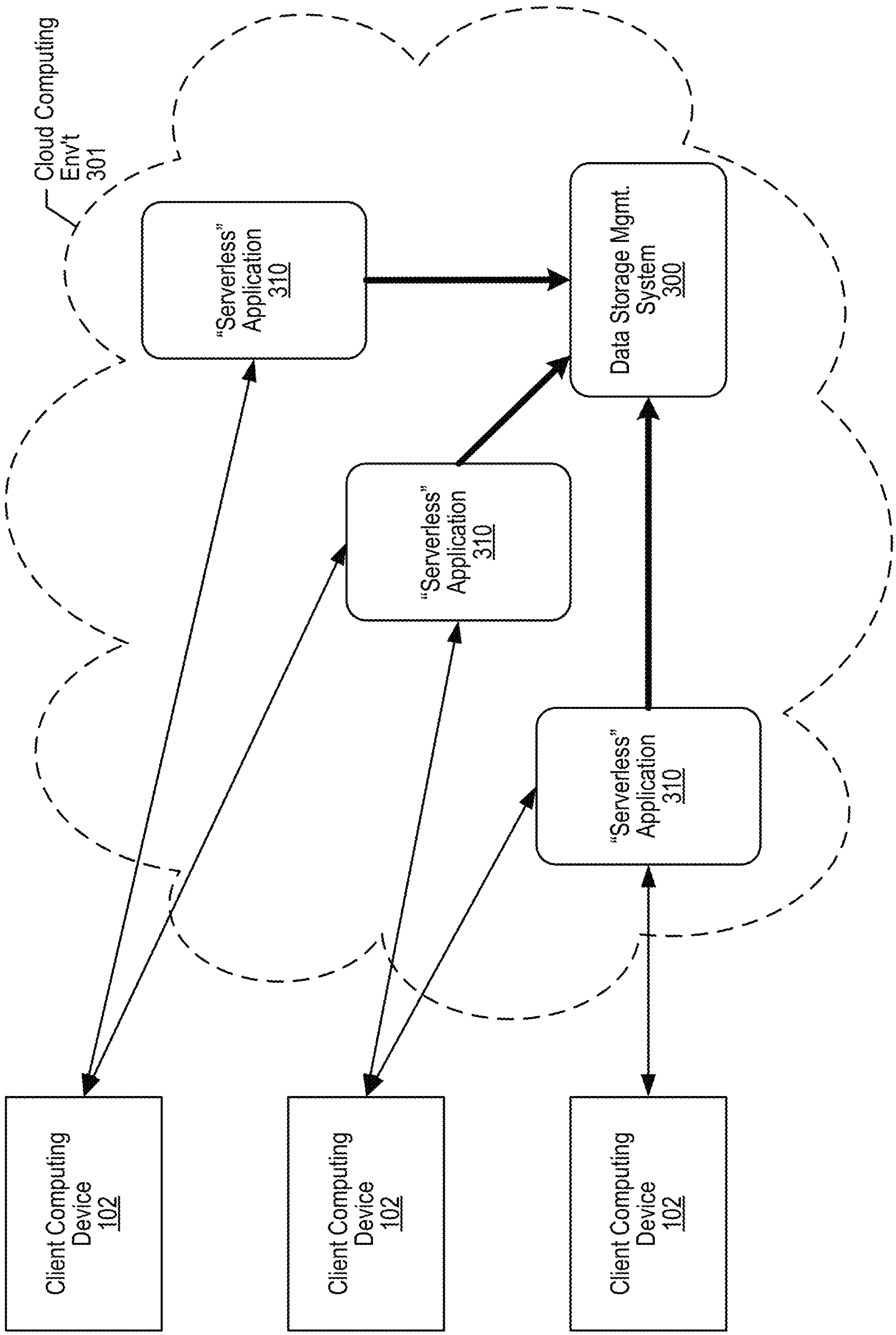


FIG. 3A System 300 For Holistically Protecting "Serverless" Applications

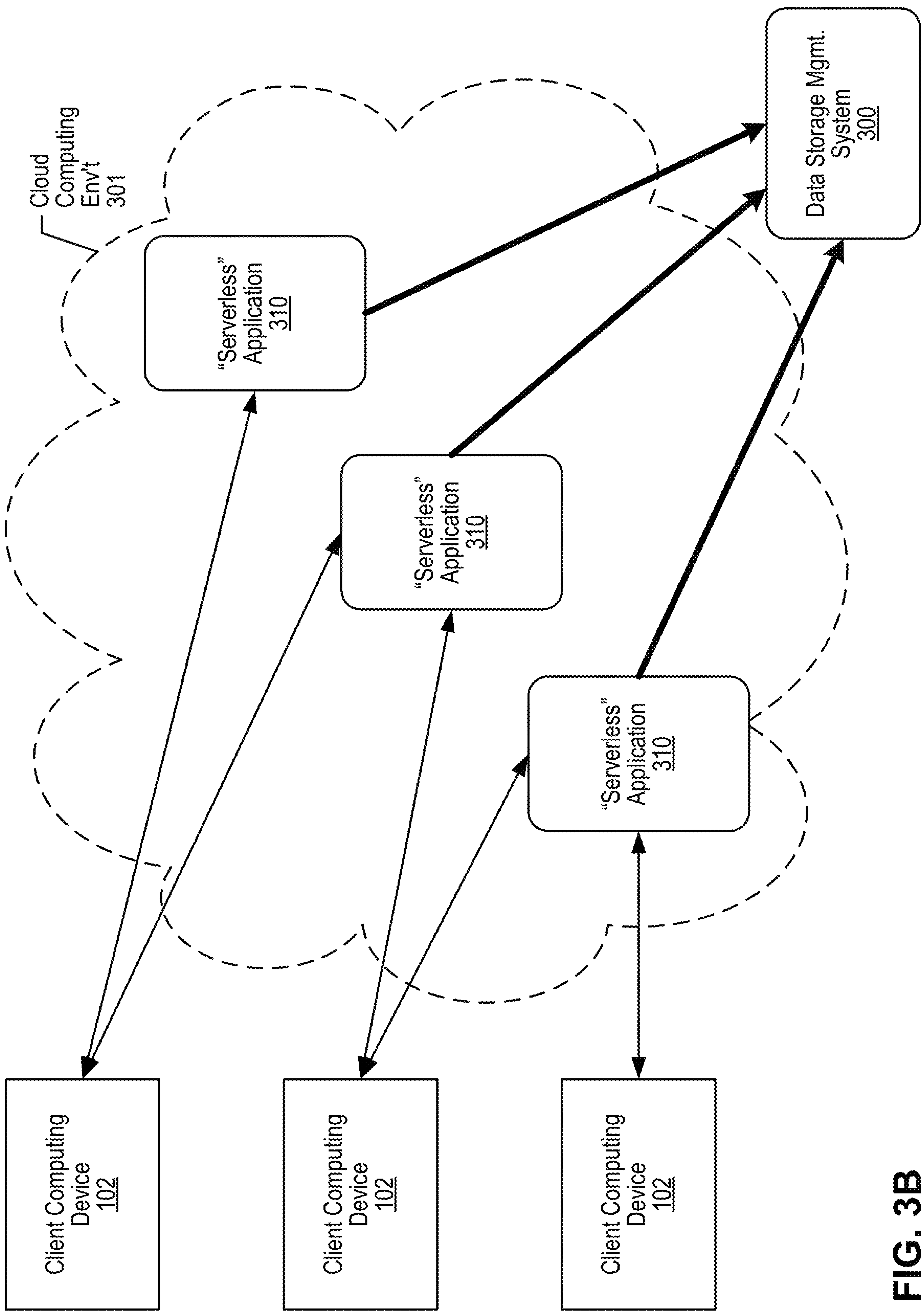

Cloud
Computing
Env't
301
Data Storage Mgmt.
System
300
"Serverless"
Application
310
"Serverless"
Application
310
"Serverless"
Application
310
Client Computing
Device
102
Client Computing
Device
102
Client Computing
Device
102

FIG. 3B

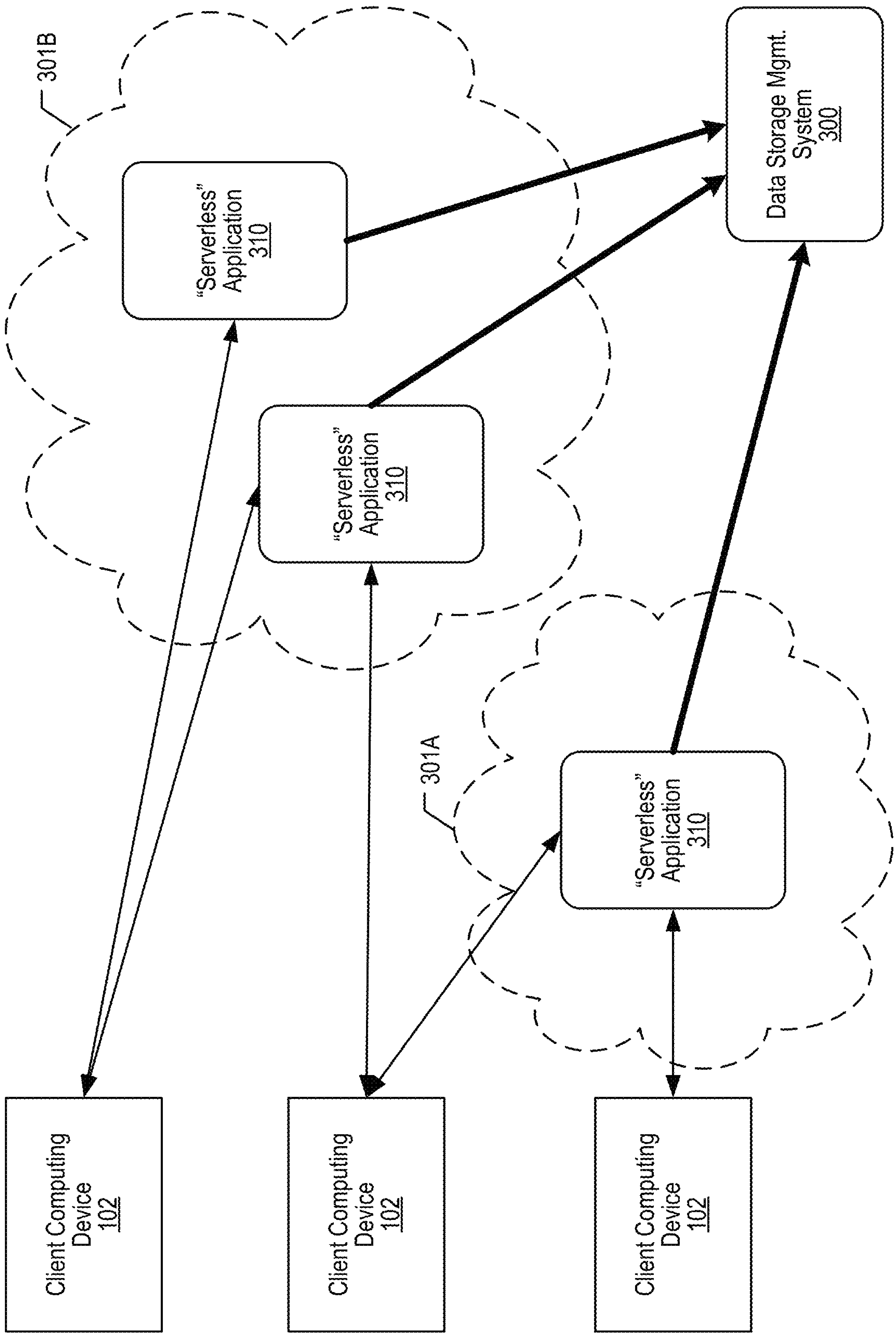

301B
"Serverless" Application 310
"Serverless" Application 310
Data Storage Mgmt. System 300
301A
"Serverless" Application 310
Client Computing Device 102
Client Computing Device 102
Client Computing Device 102

FIG. 3C

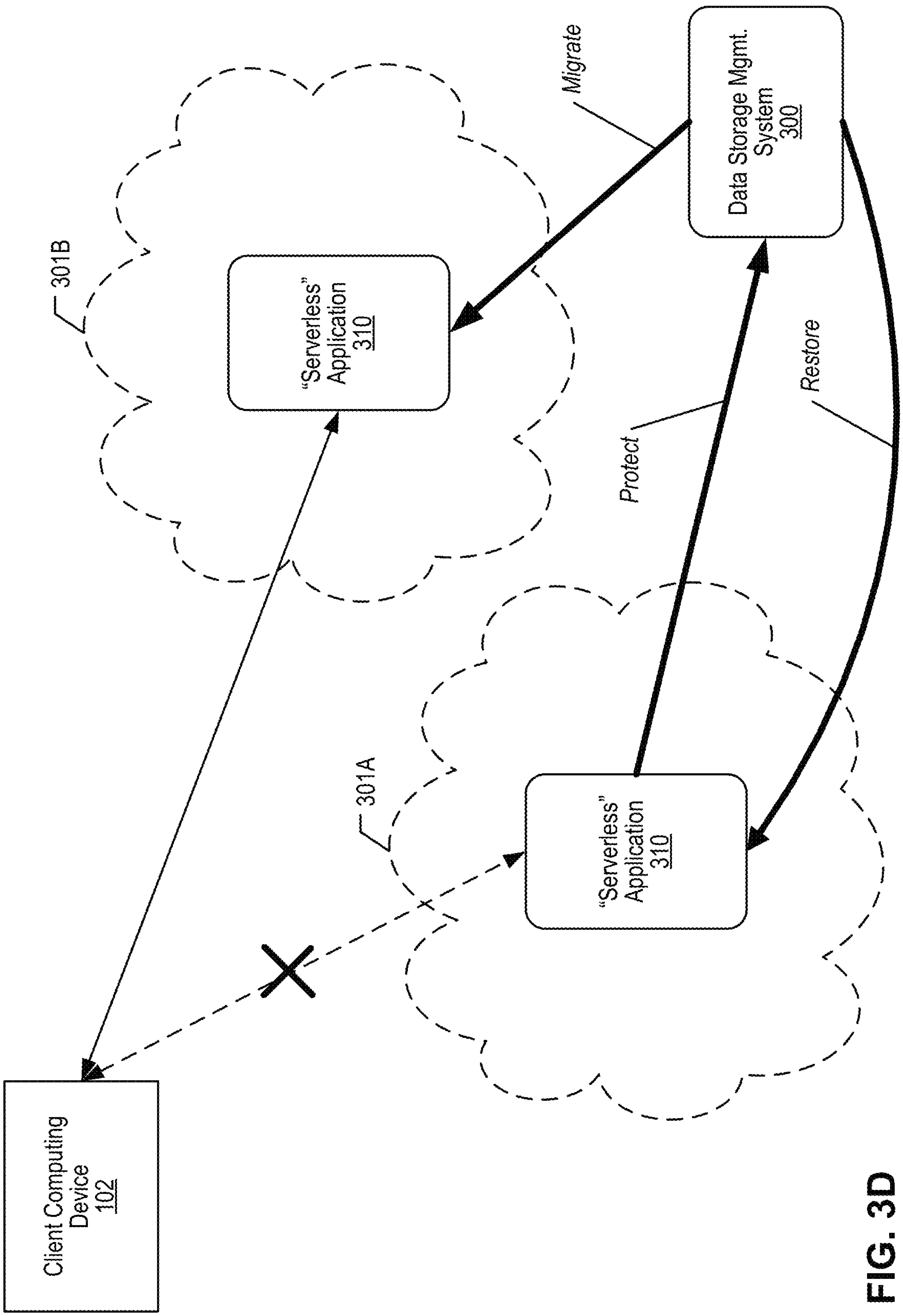

301B
"Serverless" Application 310
Migrate
Data Storage Mgmt. System 300
Protect
Restore
301A
"Serverless" Application 310
Client Computing Device 102

FIG. 3D

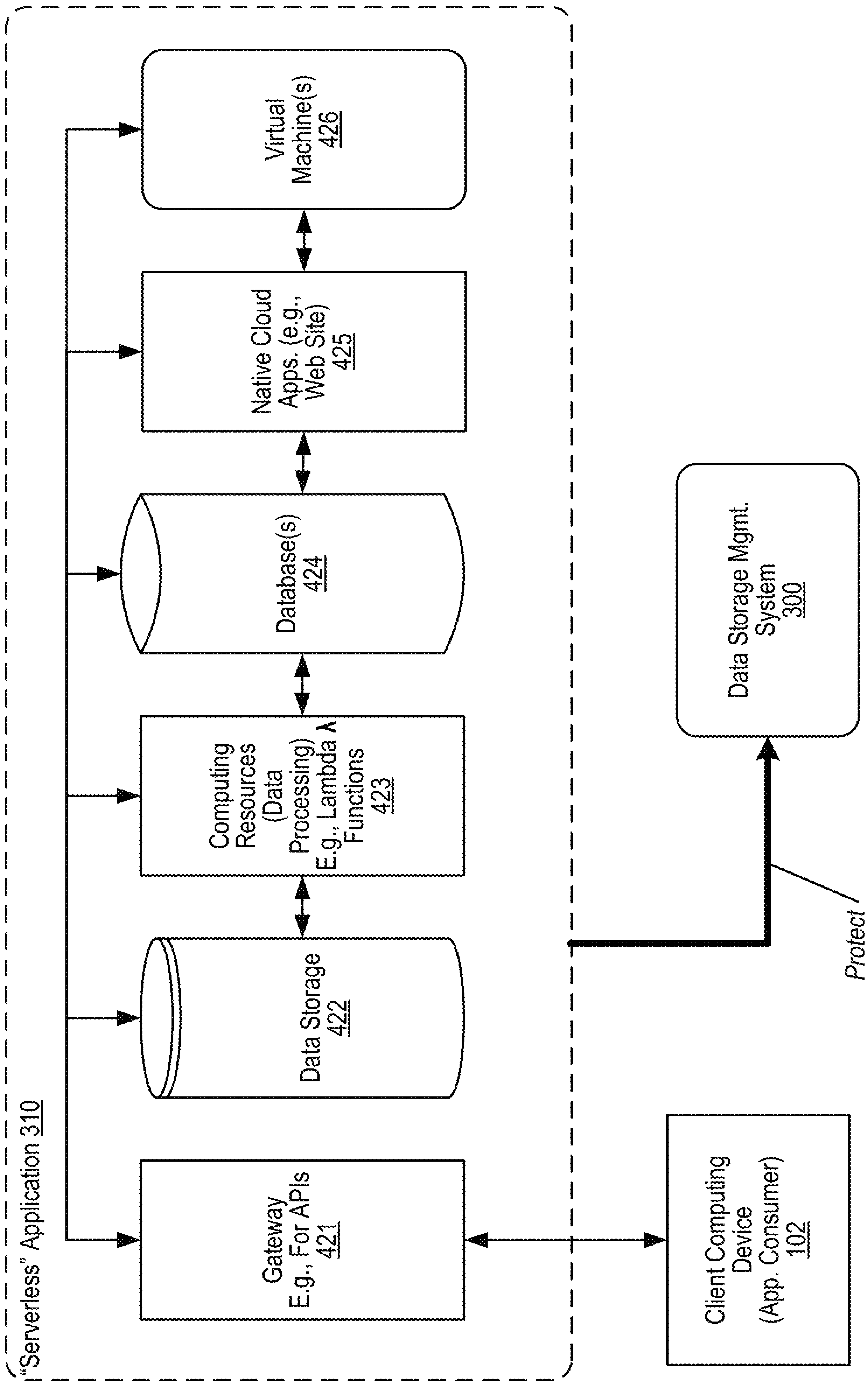

"Serverless" Application 310
Gateway
E.g., For APIs
421
Data Storage
422
Computing
Resources
(Data
Processing)
E.g., Lambda λ
Functions
423
Database(s)
424
Native Cloud
Apps. (e.g.,
Web Site)
425
Virtual
Machine(s)
426
Data Storage Mgmt.
System
300
Protect
Client Computing
Device
(App. Consumer)
102

FIG. 4

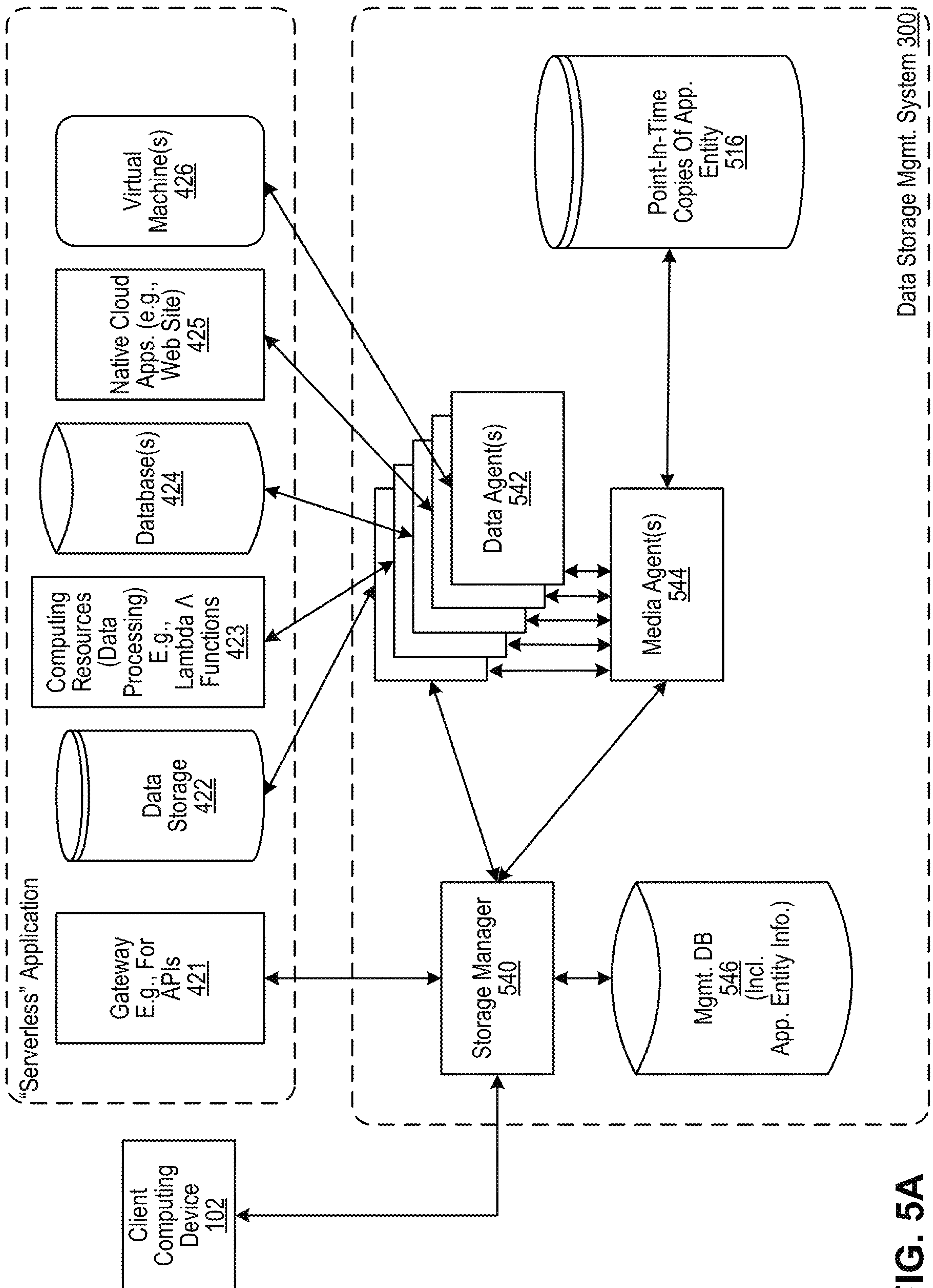

"Serverless" Application
Gateway E.g., For APIs 421
Data Storage 422
Computing Resources (Data Processing) E.g., Lambda Λ Functions 423
Database(s) 424
Native Cloud Apps. (e.g., Web Site) 425
Virtual Machine(s) 426
Data Storage Mgmt. System 300
Point-In-Time Copies Of App. Entity 516
Data Agent(s) 542
Media Agent(s) 544
Storage Manager 540
Mgmt. DB 546 (Incl. App. Entity Info.)
Client Computing Device 102

FIG. 5A

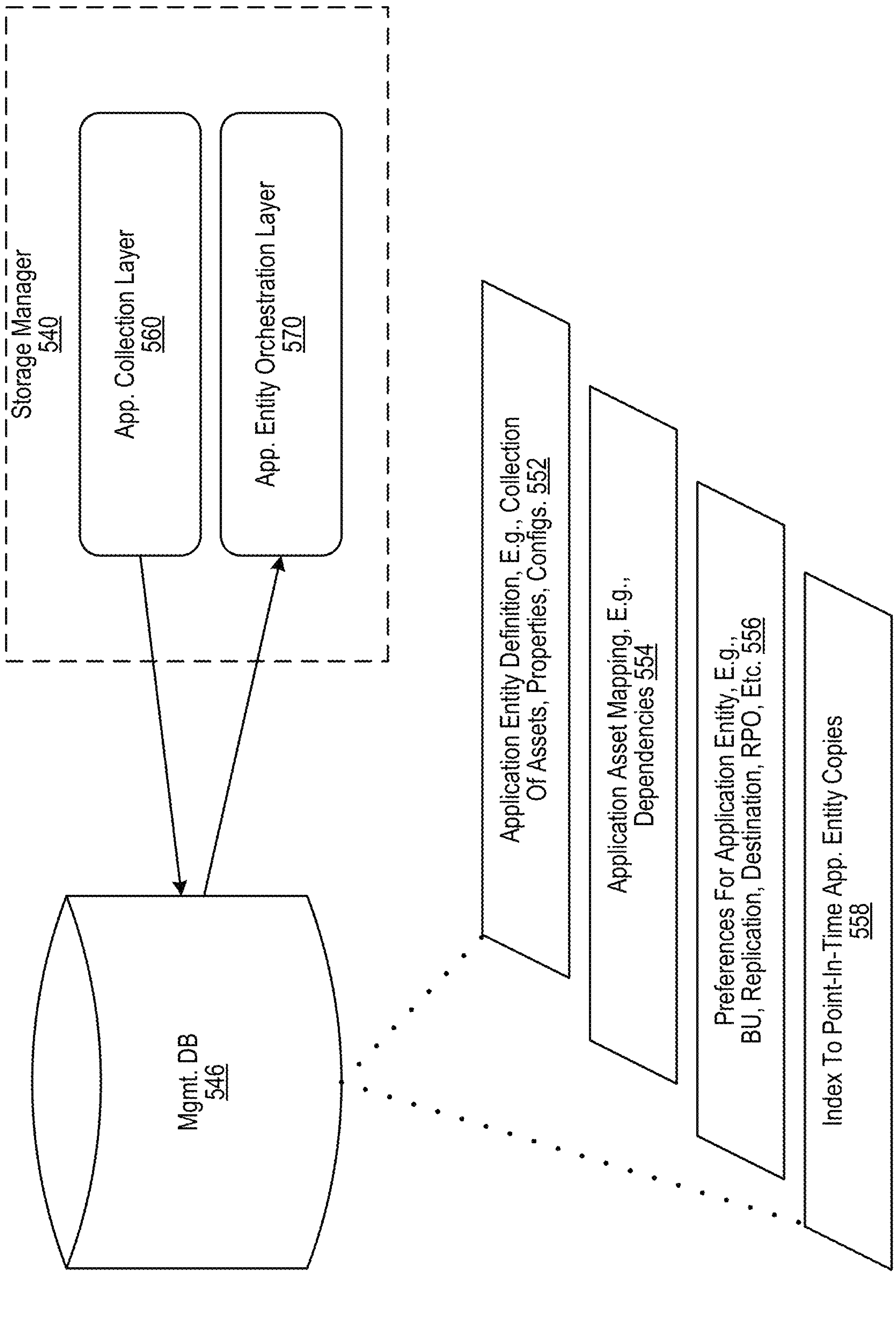

Storage Manager
540
App. Collection Layer
560
App. Entity Orchestration Layer
570
Mgmt. DB
546
Application Entity Definition, E.g., Collection Of Assets, Properties, Configs. 552
Application Asset Mapping, E.g., Dependencies 554
Preferences For Application Entity, E.g., BU, Replication, Destination, RPO, Etc. 556
Index To Point-In-Time App. Entity Copies
558

FIG. 5B

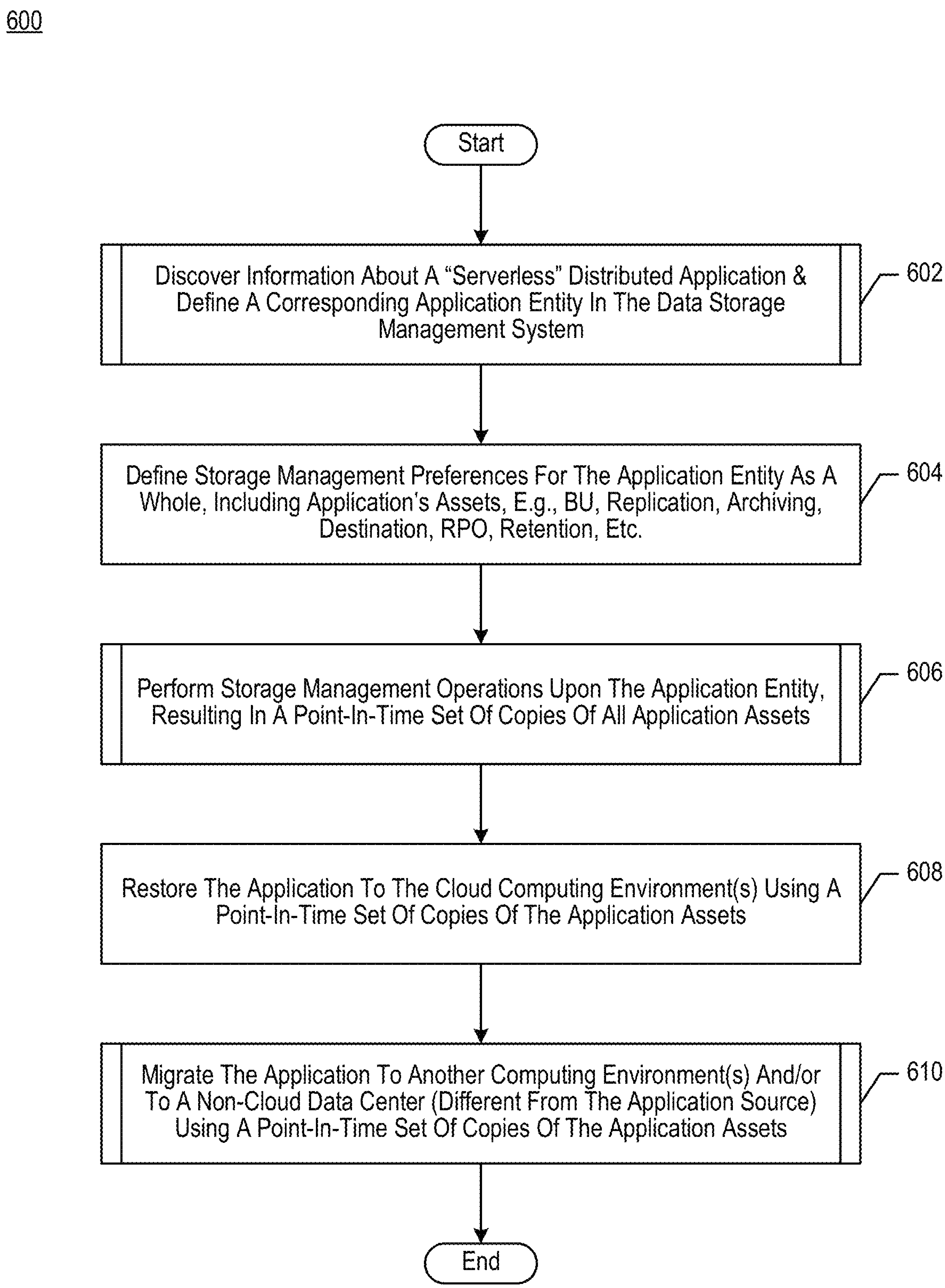

600
Start
Discover Information About A "Serverless" Distributed Application & Define A Corresponding Application Entity In The Data Storage Management System
602
Define Storage Management Preferences For The Application Entity As A Whole, Including Application's Assets, E.g., BU, Replication, Archiving, Destination, RPO, Retention, Etc.
604
Perform Storage Management Operations Upon The Application Entity, Resulting In A Point-In-Time Set Of Copies Of All Application Assets
606
Restore The Application To The Cloud Computing Environment(s) Using A Point-In-Time Set Of Copies Of The Application Assets
608
Migrate The Application To Another Computing Environment(s) And/or To A Non-Cloud Data Center (Different From The Application Source) Using A Point-In-Time Set Of Copies Of The Application Assets
610
End

FIG. 6

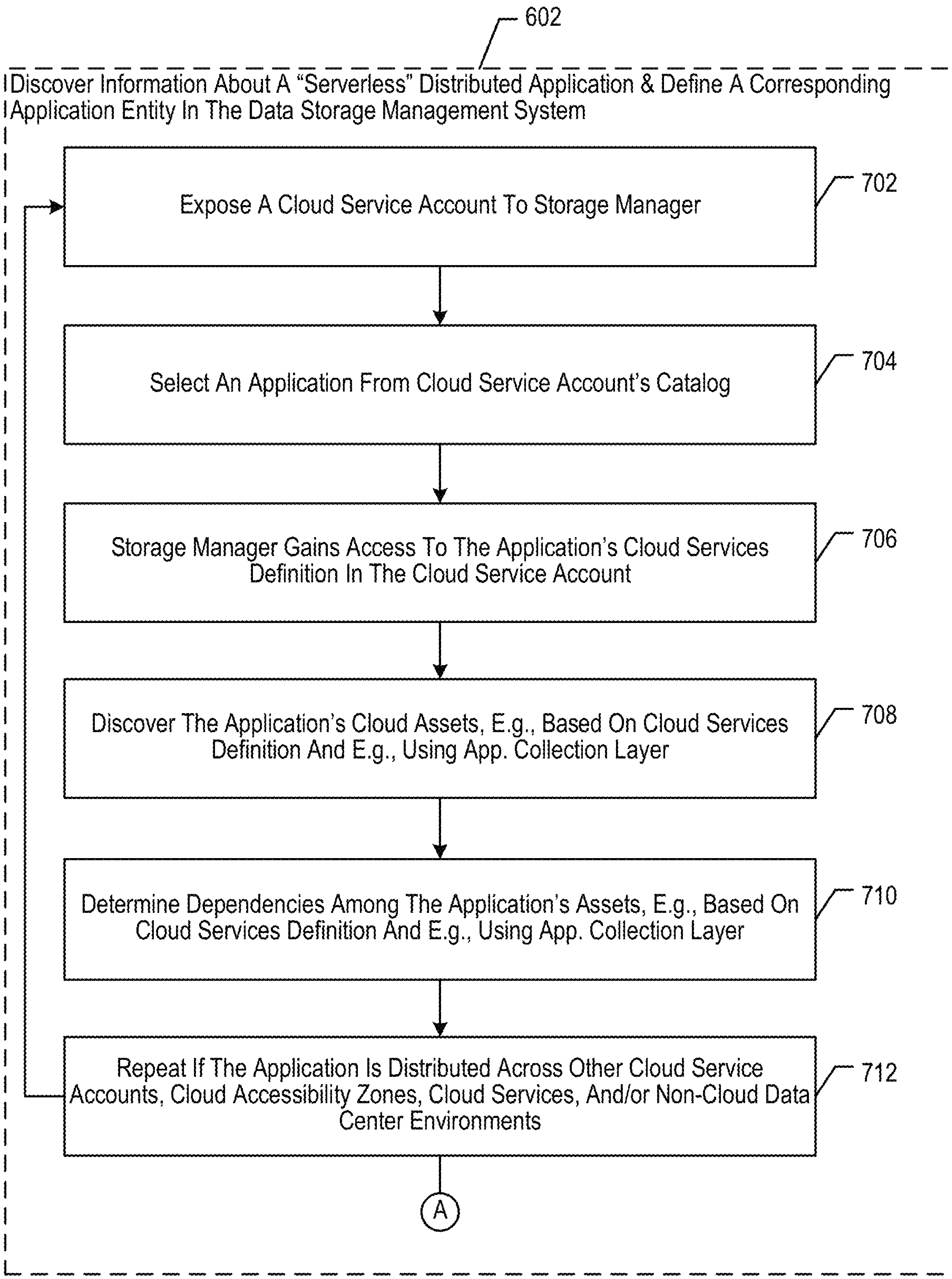

602
Discover Information About A "Serverless" Distributed Application & Define A Corresponding Application Entity In The Data Storage Management System
Expose A Cloud Service Account To Storage Manager
702
Select An Application From Cloud Service Account's Catalog
704
Storage Manager Gains Access To The Application's Cloud Services Definition In The Cloud Service Account
706
Discover The Application's Cloud Assets, E.g., Based On Cloud Services Definition And E.g., Using App. Collection Layer
708
Determine Dependencies Among The Application's Assets, E.g., Based On Cloud Services Definition And E.g., Using App. Collection Layer
710
Repeat If The Application Is Distributed Across Other Cloud Service Accounts, Cloud Accessibility Zones, Cloud Services, And/or Non-Cloud Data Center Environments
712
A

FIG. 7A

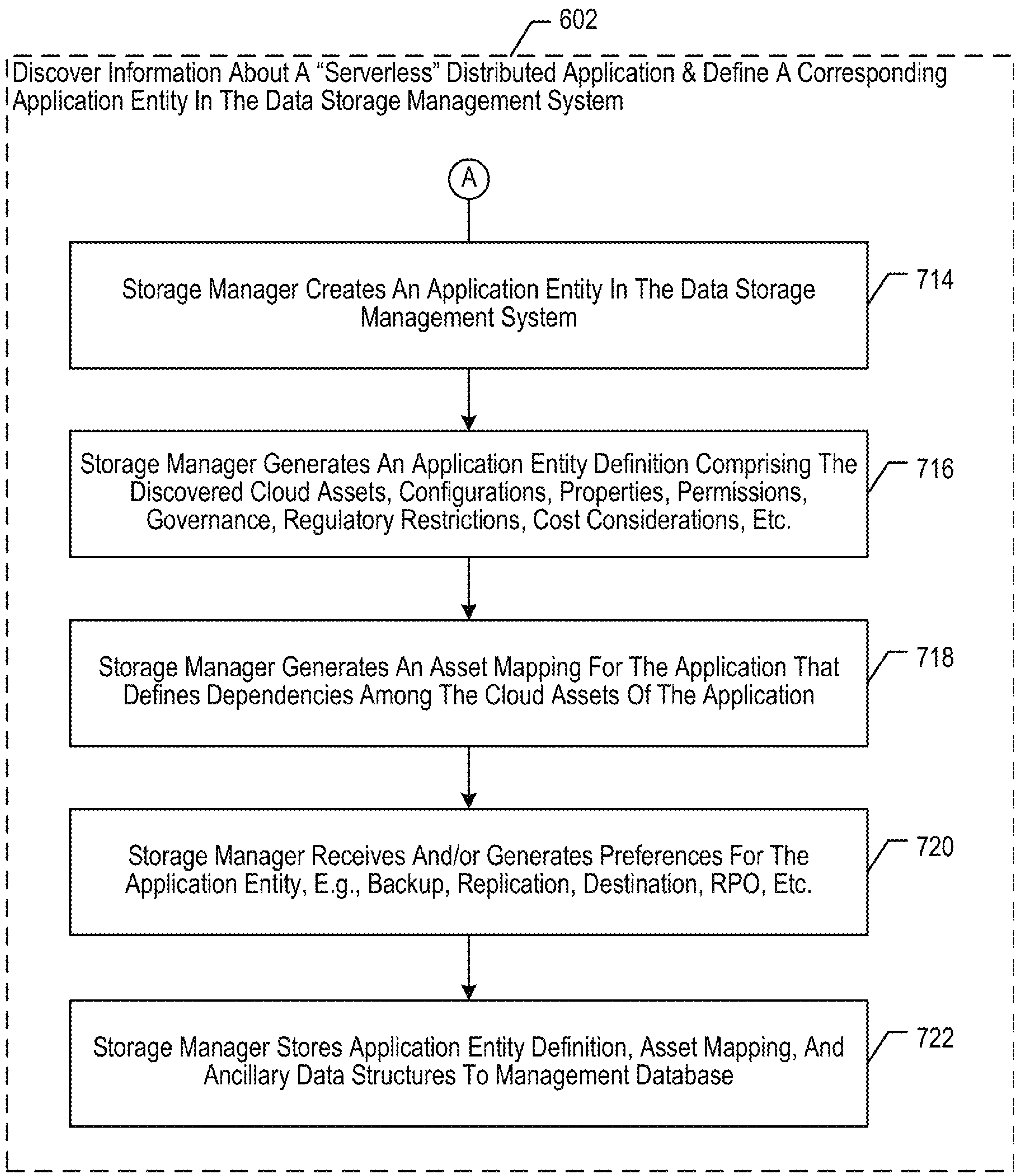

602
Discover Information About A "Serverless" Distributed Application & Define A Corresponding Application Entity In The Data Storage Management System
A
Storage Manager Creates An Application Entity In The Data Storage Management System
714
Storage Manager Generates An Application Entity Definition Comprising The Discovered Cloud Assets, Configurations, Properties, Permissions, Governance, Regulatory Restrictions, Cost Considerations, Etc.
716
Storage Manager Generates An Asset Mapping For The Application That Defines Dependencies Among The Cloud Assets Of The Application
718
Storage Manager Receives And/or Generates Preferences For The Application Entity, E.g., Backup, Replication, Destination, RPO, Etc.
720
Storage Manager Stores Application Entity Definition, Asset Mapping, And Ancillary Data Structures To Management Database
722

FIG. 7B

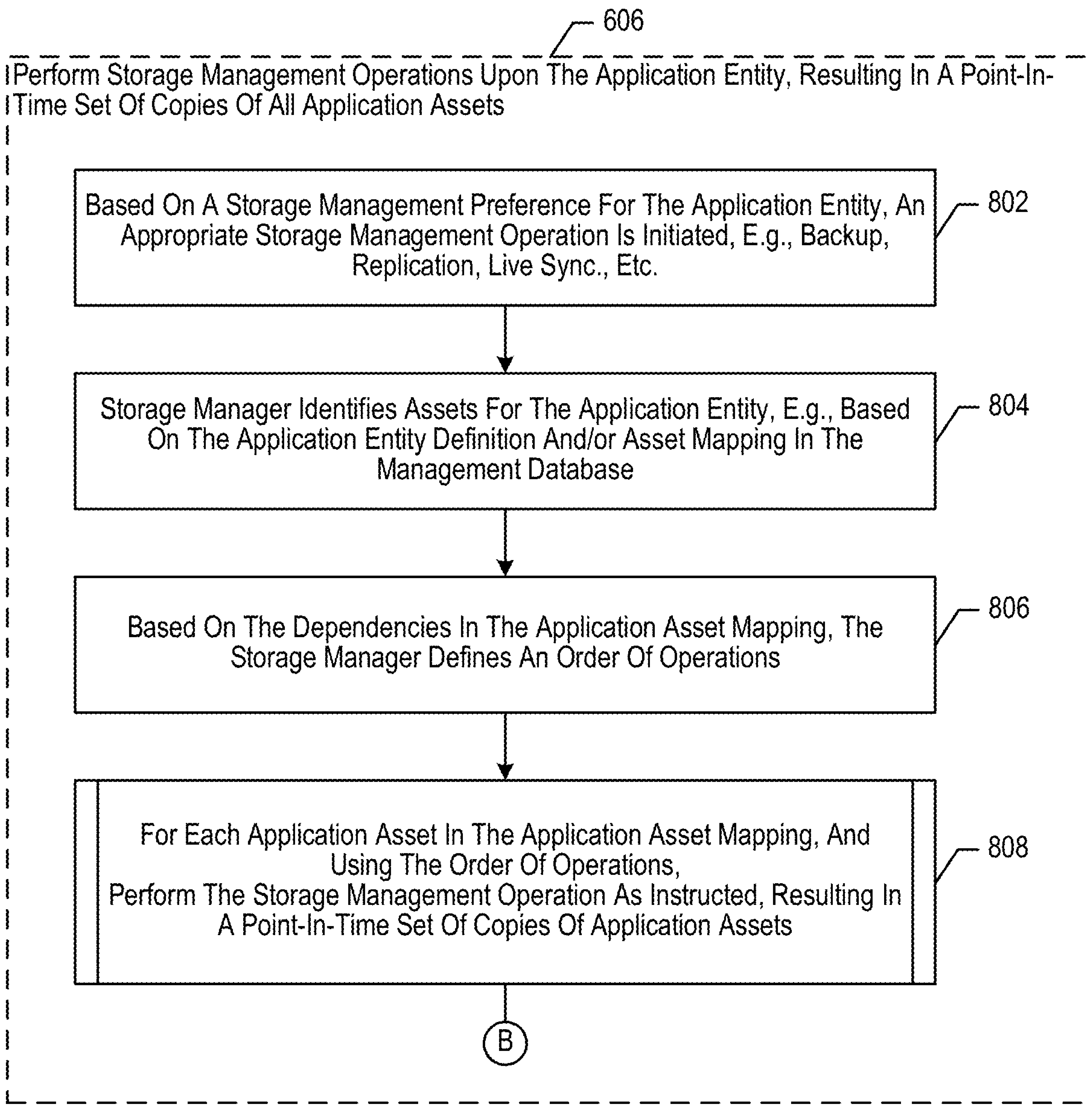

606
Perform Storage Management Operations Upon The Application Entity, Resulting In A Point-In-Time Set Of Copies Of All Application Assets
Based On A Storage Management Preference For The Application Entity, An Appropriate Storage Management Operation Is Initiated, E.g., Backup, Replication, Live Sync., Etc.
802
Storage Manager Identifies Assets For The Application Entity, E.g., Based On The Application Entity Definition And/or Asset Mapping In The Management Database
804
Based On The Dependencies In The Application Asset Mapping, The Storage Manager Defines An Order Of Operations
806
For Each Application Asset In The Application Asset Mapping, And Using The Order Of Operations, Perform The Storage Management Operation As Instructed, Resulting In A Point-In-Time Set Of Copies Of Application Assets
808
B

FIG. 8a

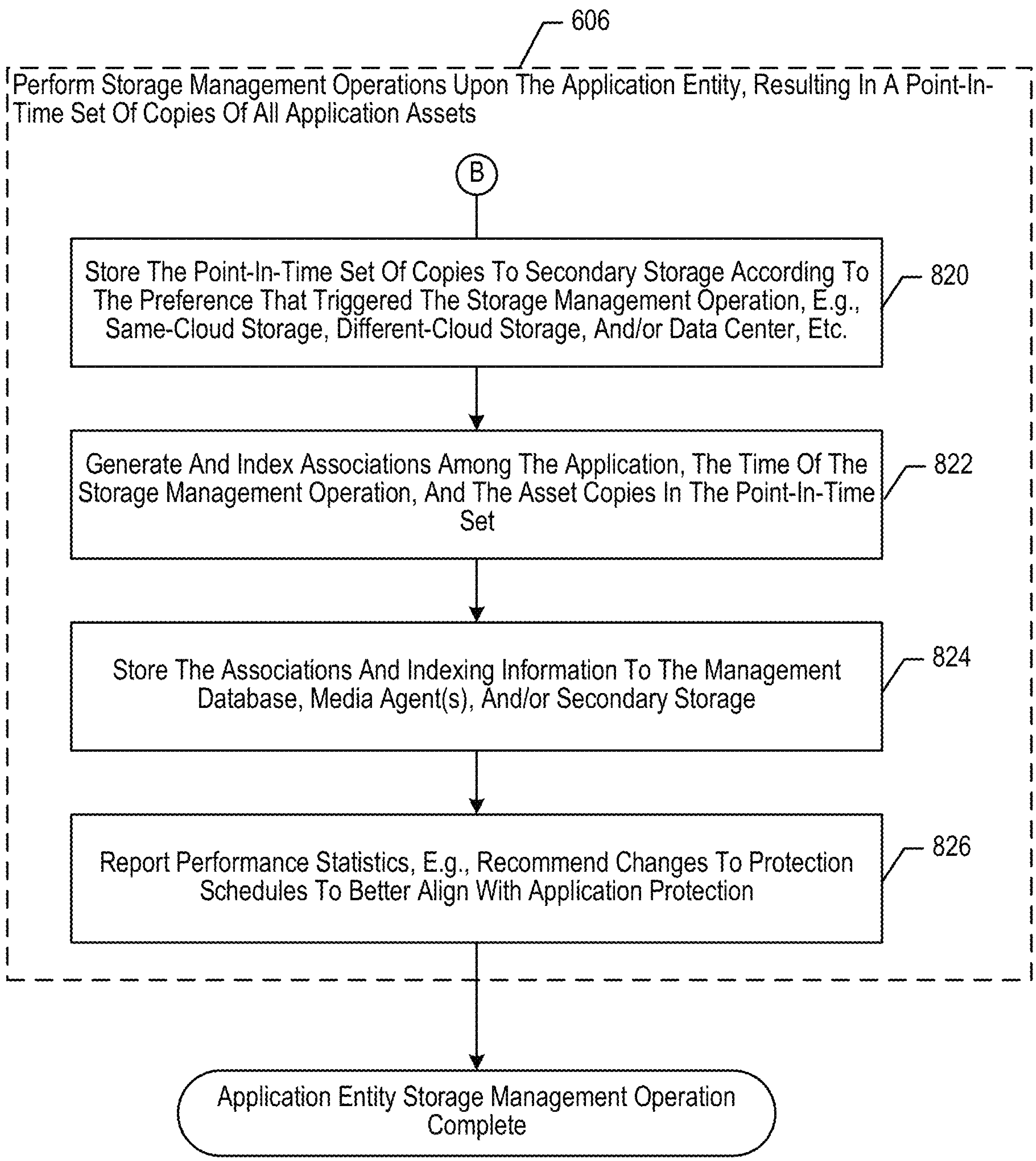

606
Perform Storage Management Operations Upon The Application Entity, Resulting In A Point-In-Time Set Of Copies Of All Application Assets
B
Store The Point-In-Time Set Of Copies To Secondary Storage According To The Preference That Triggered The Storage Management Operation, E.g., Same-Cloud Storage, Different-Cloud Storage, And/or Data Center, Etc.
820
Generate And Index Associations Among The Application, The Time Of The Storage Management Operation, And The Asset Copies In The Point-In-Time Set
822
Store The Associations And Indexing Information To The Management Database, Media Agent(s), And/or Secondary Storage
824
Report Performance Statistics, E.g., Recommend Changes To Protection Schedules To Better Align With Application Protection
826
Application Entity Storage Management Operation Complete

FIG. 8B

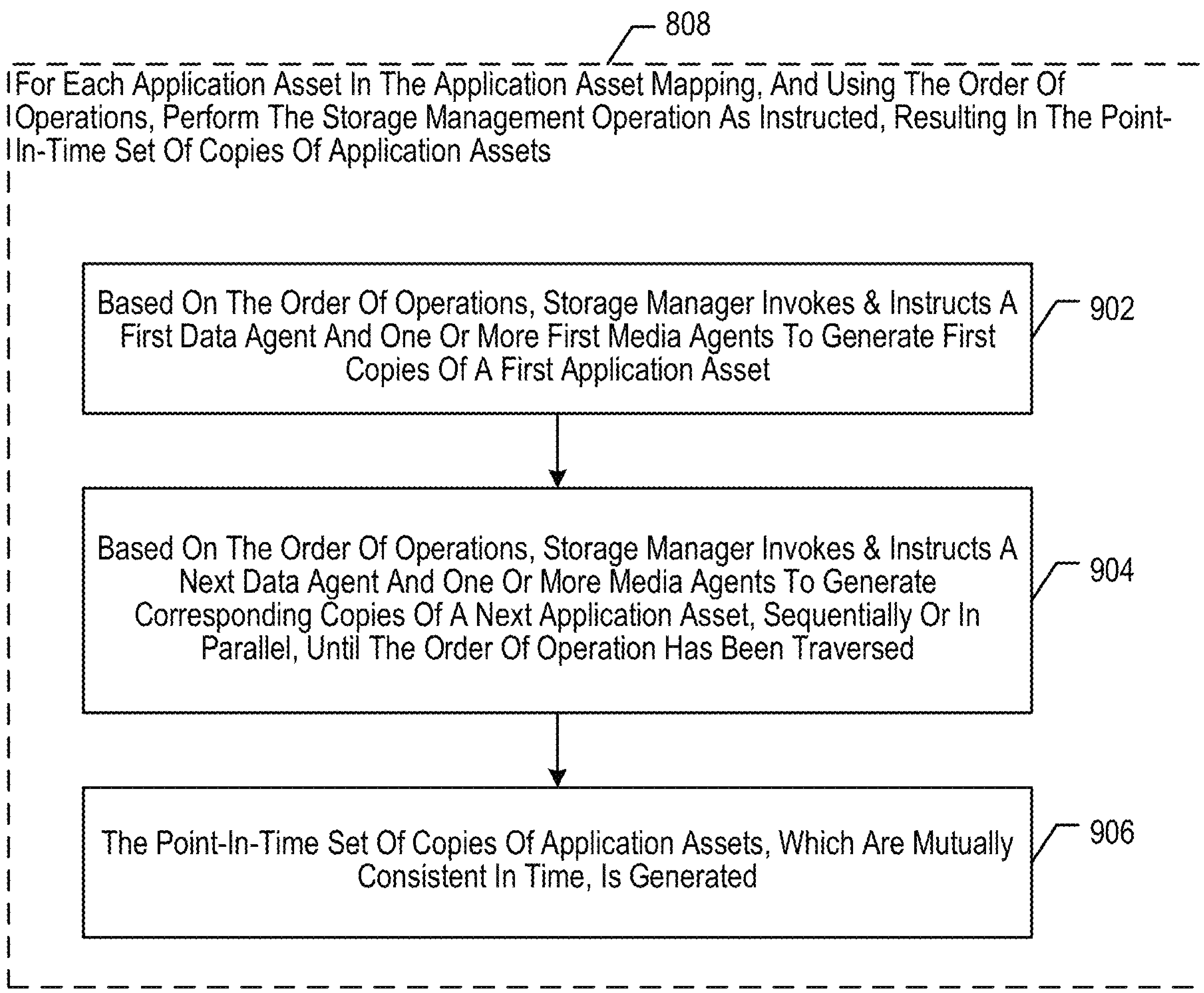

808
For Each Application Asset In The Application Asset Mapping, And Using The Order Of Operations, Perform The Storage Management Operation As Instructed, Resulting In The Point-In-Time Set Of Copies Of Application Assets
Based On The Order Of Operations, Storage Manager Invokes & Instructs A First Data Agent And One Or More First Media Agents To Generate First Copies Of A First Application Asset
902
Based On The Order Of Operations, Storage Manager Invokes & Instructs A Next Data Agent And One Or More Media Agents To Generate Corresponding Copies Of A Next Application Asset, Sequentially Or In Parallel, Until The Order Of Operation Has Been Traversed
904
The Point-In-Time Set Of Copies Of Application Assets, Which Are Mutually Consistent In Time, Is Generated
906

FIG. 9

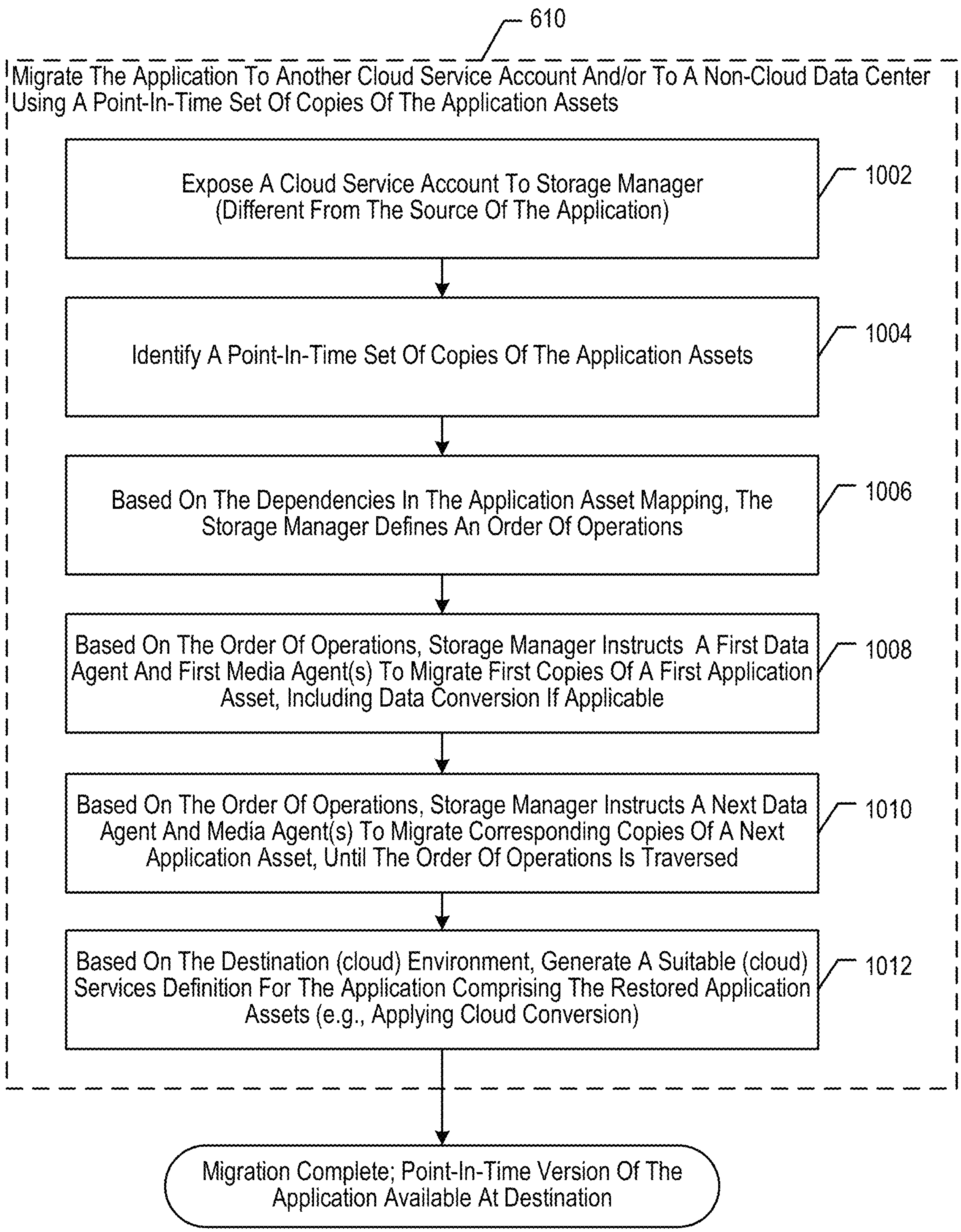

610
Migrate The Application To Another Cloud Service Account And/or To A Non-Cloud Data Center Using A Point-In-Time Set Of Copies Of The Application Assets
Expose A Cloud Service Account To Storage Manager (Different From The Source Of The Application)
1002
Identify A Point-In-Time Set Of Copies Of The Application Assets
1004
Based On The Dependencies In The Application Asset Mapping, The Storage Manager Defines An Order Of Operations
1006
Based On The Order Of Operations, Storage Manager Instructs A First Data Agent And First Media Agent(s) To Migrate First Copies Of A First Application Asset, Including Data Conversion If Applicable
1008
Based On The Order Of Operations, Storage Manager Instructs A Next Data Agent And Media Agent(s) To Migrate Corresponding Copies Of A Next Application Asset, Until The Order Of Operations Is Traversed
1010
Based On The Destination (cloud) Environment, Generate A Suitable (cloud) Services Definition For The Application Comprising The Restored Application Assets (e.g., Applying Cloud Conversion)
1012
Migration Complete; Point-In-Time Version Of The Application Available At Destination

FIG. 10

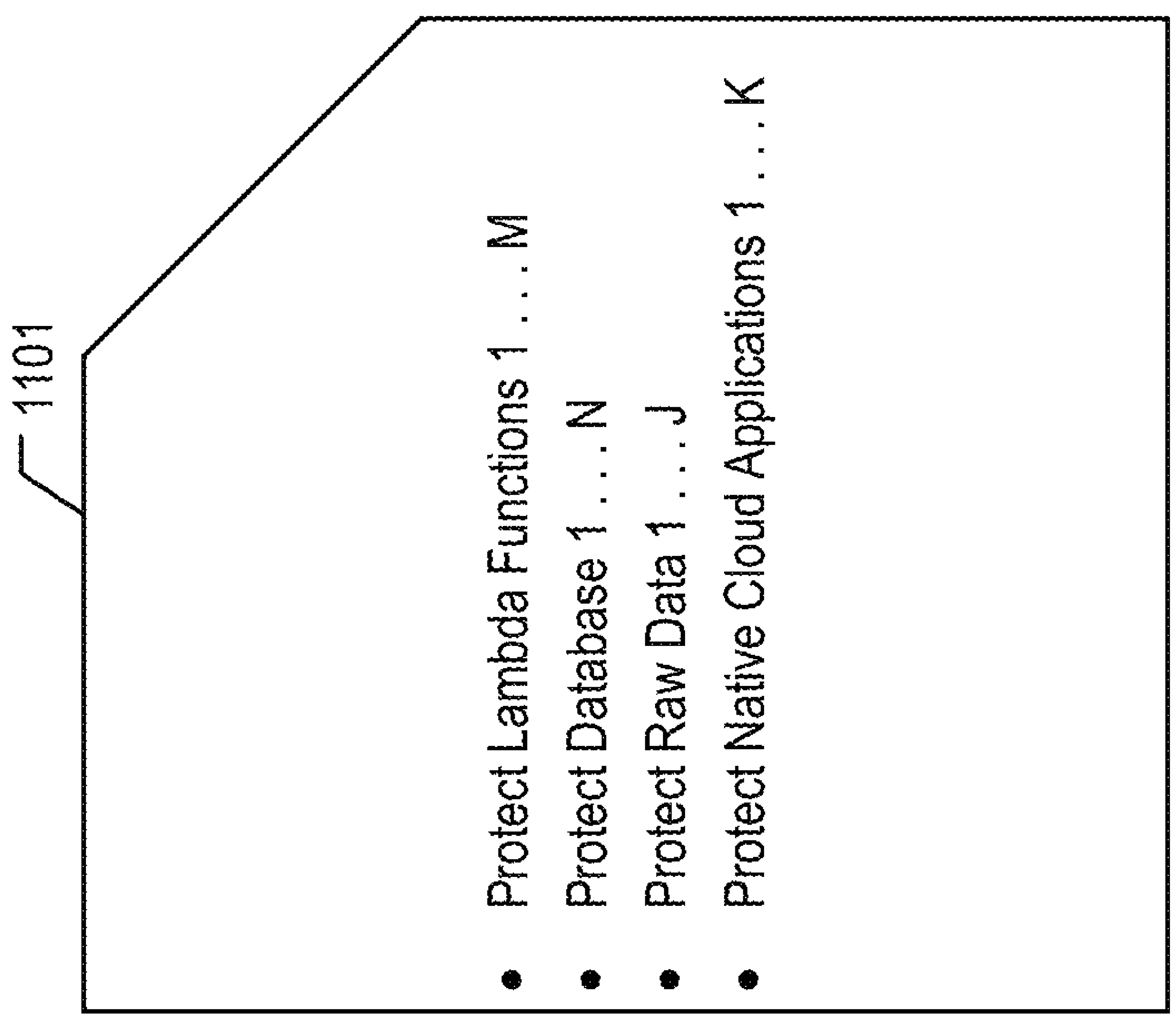


FIG. 11 Illustrative Order Of Operations

1201

- Running Virtual Computing Environments, E.g., Instances
- Pre-Configured Templates For The Virtual Computing Environments
- Secure Login Information, E.g., Active Directory
- Storage For Temporary Data That Is Deleted When The Virtual Computing Environment Ends, E.g., Instance Store Volumes
- Persistent Storage, E.g., Amazon Elastic Block Store (EBS)
- Metadata To Assign To The Cloud Service Account's Resources
- Virtual Networks That Are Isolated From The Rest Of The Cloud Computing Environment, E.g., Virtual Private Clouds
- Snapshots
- Security Groups
- Application Configurations, E.g., CPU Setting, Firewall, IP Address, Load Balancer, Etc.

FIG. 12 Illustrative Cloud Computing Environment Assets Configured In A Cloud Service Account

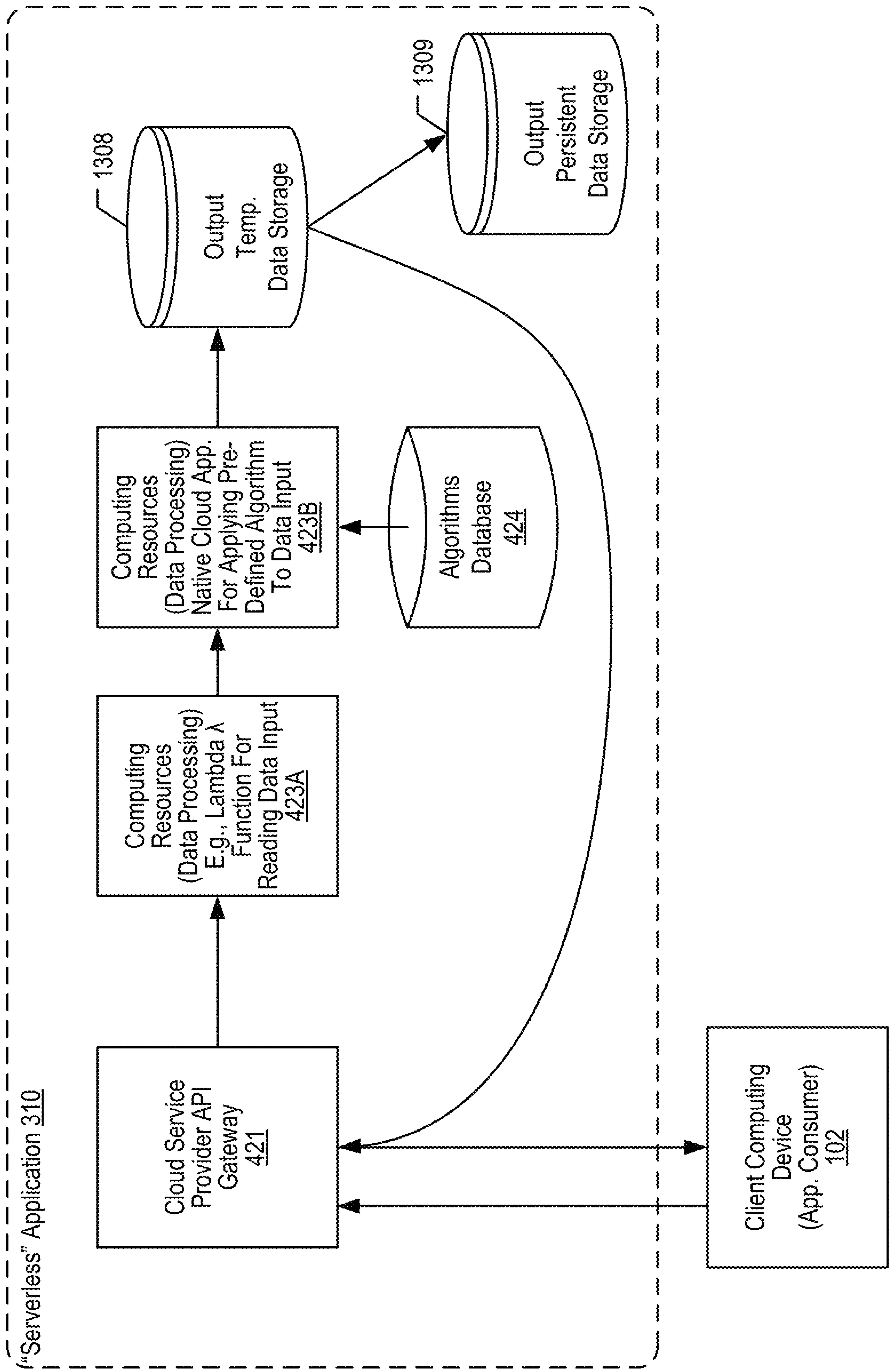


FIG. 13 Illustrative "Serverless" Distributed Application Service Definition In A Cloud Computing Environment

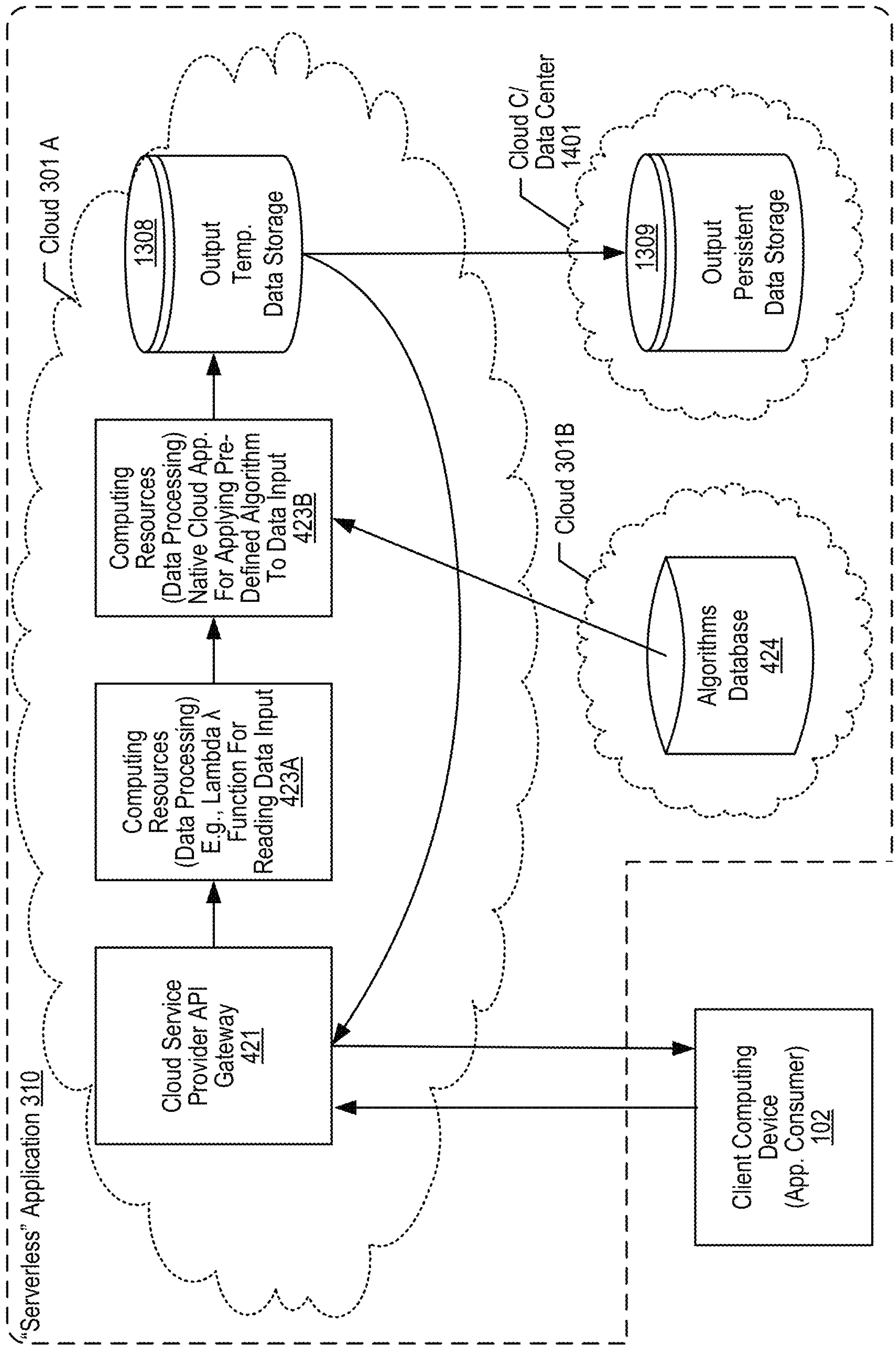


FIG. 14 Illustrative Service Definition Of A "Serverless" Application Distributed Across Multiple Cloud Computing Environments

DATA STORAGE MANAGEMENT SYSTEM FOR HOLISTIC PROTECTION AND MIGRATION OF SERVERLESS APPLICATIONS ACROSS MULTI-CLOUD COMPUTING ENVIRONMENTS

INCORPORATION BY REFERENCE TO ANY PRIORITY APPLICATIONS

This application claims the benefit of priority to U.S. Provisional Patent Application Ser. No. 62/841,126 filed on Apr. 30, 2019. Any and all applications for which a foreign or domestic priority claim is identified in the Application Data Sheet of the present application are hereby incorporated by reference in their entireties under 37 CFR 1.57.

COPYRIGHT NOTICE

BACKGROUND

Businesses recognize the commercial value of their data and seek reliable, cost-effective ways to protect the information stored on their computer networks while minimizing impact on productivity. A company might back up critical computing systems such as databases, file servers, web servers, virtual machines, and so on as part of a planned maintenance schedule. As cloud computing becomes ubiquitous in enterprise data processing, protecting the applications that run in cloud computing environments and the applications' associated data poses a number of technological challenges, whether in single-cloud or multi-cloud scenarios.

SUMMARY

The present inventors devised an approach to protecting, restoring, and migrating so-called "serverless" applications that are deployed within one or more cloud computing environments, which are sometimes referred to as "cloud-native distributed applications." A "cloud computing environment" as used herein comprises a collection (or suite) of resources provided as a service by a cloud service provider to a cloud service account. A cloud computing environment is accessed via the cloud service account, which entitles the subscriber to a suite of services in a particular cloud service supplied by a cloud service provider. Cloud computing environments vary among cloud services, among cloud availability zones, and even among cloud service accounts from the same cloud service provider. A cloud computing environment as used herein need not comprise data processing (computing) resources and can be limited to data storage and retrieval features.

Serverless applications use various resources that are distributed within a cloud computing environment, across cloud availability zones, and/or across multiple cloud computing environments. Typically, serverless applications are hosted in a cloud service and use transitory, temporary, and/or persistent cloud storage to store their data. Serverless applications are enabled by cloud infrastructure that eliminates the need for application creators to closely manage the infrastructure needed for the application, such as provisioning servers, clusters, virtual machines, storage devices, and/or network resources. Instead, an application's creator uses resources made available in the cloud computing environment to construct the application, run the application, and store relevant data for the application, including databases of information and/or results generated by the application. Thus, serverless applications can be referred to as distributed applications. Additionally, applications with distributed assets that do not meet the definition of "serverless" also are referred to herein as "distributed applications," e.g., applications that operate within non-cloud data centers.

Examples of serverless applications include web-based applications. A weather service accessible to consumers provides a useful example, without limitation. In such an exemplary serverless application, geographic and long-term climate databases are stored in the cloud. As new weather data is fed to a weather analysis function operating in the cloud, the application generates weather forecasts and stores the forecast information in the cloud. Consumers consume the weather forecasts when they access the weather application. In this example, some of the application's functions are supplied by the application's creator, e.g., weather data processing and analysis. Other functions are supplied by the cloud service provider, e.g., application triggers, statistical analysis features, etc. Data storage associated with the weather application includes the weather forecasts available to consumers as well as databases comprising geographic and long-term climate data.

The resources used by the exemplary weather application, and more generally, resources used by other serverless/distributed applications, are referred to herein as "assets" of the application. Thus, an application's assets are resources used in whole or in part by the application to perform its function. Some assets are cloud-based, but not necessarily all. In the weather forecast example above, the application's creators supply the weather analysis programming (hosted in the cloud), specify which cloud-based databases are to provide geographic and long-term climate data, and define how the resultant weather forecasts are to be stored in cloud storage and presented to application users. The cloud computing environment supplies the data processing (computing) resources, statistical analysis features, and associated data storage without necessarily requiring the application's creators to request them, expressly set them up, and/or maintain them.

In cloud computing, a serverless application is characterized by a "cloud services definition." The cloud services definition specifies the resources supplied by the application's creator as well as other resources needed by the application and supplied by the cloud computing environment. The cloud services definition comprises necessary building blocks for the application as well as relationships among the building blocks, such as data sources, data destinations, functional blocks, triggers, dependencies, etc. Thus, a cloud services definition represents the application's architecture in terms of cloud services assets. Examples of cloud services assets include without limitation: running virtual computing environments (e.g., instances); pre-configured templates for the virtual computing environments; secure login information (e.g., based on Active Directory); storage for temporary data (e.g., instance storage volumes); persistent data storage; metadata to assign to the cloud account's resources; virtual networks; snapshots; security groups; application configurations such as CPU settings, firewall information, IP addresses, load balancer resources, etc.; lambda functions; native cloud applications; virtual machines; databases; block storage; file storage; etc., without limitation. Nomenclatures and functionality for cloud services assets vary among cloud service providers and implementations; likewise, how a cloud services definition is created, expressed, and presented to users also varies among cloud service providers; however, based at least in part on documentation supplied by the cloud service provider, the vendor-specific nomenclature and functionality will be clear to those skilled in the art.

One of the features of the illustrative data storage management system is to enable portability and migration of serverless applications from a source environment to a destination environment that is substantially different from the source. Accordingly, the system is configured to recognize how a given asset at the source might differ from an equivalent asset at the destination and to perform suitable conversions to successfully complete migration from one computing environment to another. For example, a given statistical feature (e.g., moving average, etc.) might be invoked differently by different cloud services. Long-term data storage might be addressed and/or accessed differently at different cloud services. The illustrative data storage management is configured to navigate these differences and provide suitable conversion, re-configuration, and/or adaptation, so that a serverless application can be protected (e.g., backed up, replicated, etc.) and migrated from one cloud service to another, from one cloud service account to another, from one cloud availability zone to another, and even between cloud and non-cloud data centers.

The weather forecasting example above is not limiting. Other serverless applications are created and owned by enterprises that buy/lease/use web services from one or more cloud service providers, such as Amazon (Amazon Web Services AWS), Microsoft (Azure), and others, and/or build and operate their own cloud computing environments. Accordingly, each serverless application has an architecture that defines how the application's assets interoperate.

One of the features of the illustrative data storage management system is to discover an application's cloud services assets. In some embodiments, the system obtains access to the application's cloud services definition as a starting point for asset discovery and extracts pertinent information therefrom. In other embodiments, the asset discovery process proceeds piecemeal, by analyzing assets' input/output relationships to other assets in a kind of chain-link or bootstrap approach. In such embodiments, once a first asset is exposed to the data storage management system, the discovery process identifies other relevant assets, e.g., by analyzing asset configurations to determine relationships to other assets. The asset discovery process thus varies from one cloud computing environment to another and further varies from one serverless application to another, depending on what assets are involved.

Some serverless applications are configured with assets that are distributed among two or more cloud computing environments and/or non-cloud data centers. For example, an application might use a database that is stored in a non-cloud data center rather than in the cloud computing environment that hosts the application's data processing. Furthermore, the application might store all its final outputs to data storage in a second cloud computing environment or perhaps in a cloud availability zone that is different from the zone hosting the application's data processing. Applications that cross cloud/zone boundaries are referred to herein as "multi-cloud" applications in recognition of the fact that a single cloud computing environment and/or cloud availability zone does not comprise all the application's assets. The illustrative asset discovery process for a multi-cloud application is generally more complex than single-cloud scenarios, because the data storage management system will require access to every cloud, zone, and/or non-cloud data center comprising application assets. This aspect advantageously enables the illustrative holistic approach to protection of serverless applications regardless of their footprint.

Traditional data storage management does not deal well with serverless applications' distributed architectures. An example is set forth next: in a traditional data management system, databases are protected under their own identities and preferences (e.g., database-specific storage policies); file data is protected separately by its own separate identity and preferences (e.g., file system-specific storage policies applicable to certain file data and/or block data storage); and virtual machines (VMs) and VM data are protected by VM-related preferences/policies. In the weather forecasting example above, the geographic and climate databases might be protected under their own database-specific schedules/policies, whereas the forecast results might be protected by another policy/schedule covering file data. This dispersed approach generally fails to create copies of the application's assets that are point-in-time coordinated. Moreover, the copies might not be associated with each other in a manner that enables all the necessary copies of an application to be readily found. Thus, protecting a distributed serverless application according to traditional approaches generally does not provide for an integrated point-in-time view of the application as a whole, including its assets. For multi-cloud serverless applications, point-in-time views are even more difficult to create, as each cloud computing environment and/or non-cloud data center comprising application assets is likely to be governed by different protection preferences and schedules, ultimately resulting in asynchronous and disassociated copies. Without a coordinated view, the application cannot be reliably restored to or migrated from a certain operational point in time. Therefore, a streamlined solution is needed to address these shortcomings in the prior art.

The present inventors devised a holistic approach for protecting serverless applications in single-cloud, multi-zone, multi-cloud, and/or non-cloud data center computing environments. An illustrative data storage management system discovers application assets and creates an "application entity" that references the various assets. Protection preferences apply to the application entity as a whole. An orchestration function in the system coordinates storage management operations (e.g., backup, replication, live synchronization, etc.). The system coordinates and generates a set of copies of the application's discovered assets, which represent a point-in-time view of the application. The set of copies can be restored and/or migrated to other computing services by the data storage management system.

Protection preferences (e.g., storage policies, schedule policies, Recovery Point Objectives (RPO), storage destinations, retention policies, etc.) apply to the application entity as a whole, ensuring that copies are coordinated to create the desired point-in-time view. The system also generates an "asset mapping" that captures dependencies gleaned during the asset discovery process. For example, a data processing function F may depend upon (a) user-input configuration parameters C and (b) additional data extracted from a database D. Illustratively function F depends on assets C and D. These relationships are stored in the asset mapping and they may affect an "order of operations," which is used when the storage management operation is initiated. For example, the order of operations might capture the user-input configuration parameters C first, followed by capturing the database D, followed by capturing the data processing function's executable files (e.g., binaries, etc.) in that order. In some embodiments, the order of operations is stored within or in association with the asset mapping. In other embodiments, the order of operations is determined on demand, when the storage management operation is initiated.

When a storage management operation (e.g., backup, replication, live synchronization, etc.) is to be applied to the application entity, an orchestration function coordinates individual operations (e.g., using the order of operations), so that data integrity is maintained. The orchestration function also coordinates restore and migration operations, including any cloud-to-cloud (or cloud to/from non-cloud data center) conversions that might be necessary among application assets to activate the application in a different computing environment. The illustrative data storage management system deploys suitable storage management components (e.g., storage manager, data agents, media agents, secondary storage resources) in executing the storage management operation. The result comprises a full set of copies of application assets as well as comprehensive indexing. For example and without limitation, a database data agent is deployed for a first database asset, and a different database agent is deployed for a different kind of database asset; a file system data agent is deployed for application configuration file(s), executable files, and/or data output files; a virtual server data agent is deployed for virtual machine assets, etc. Media agents are deployed for generating copies, storing copies to secondary storage, and indexing the set of copies. The full set of copies are associated with each other and the set represents a point-in-time view of the application at the time the storage operation was initiated.

In some embodiments, a given cloud services asset is associated with more than one serverless application. For example, a database might comprise data used by several serverless applications. Accordingly, the database is part of the cloud services assets for each serverless application and the database is backed up along with a given application's assets according to the preferences for the given application-entity configured in the data storage management system. Depending on different preferences for the various applications, several backups of the same database might be taken, but each backup is associated with a distinct point-in-time of a different serverless application. This approach advantageously provides each serverless application with a complete point-in-time view of its own assets regardless of the role played by those assets within another application.

The full set of copies can be restored to the original computing environment (single cloud, multi-cloud, and/or non-cloud data center) by the illustrative data storage management system to enable the application to run from the coordinated point-in-time copies. Alternatively, the application can be migrated to a different computing environment by migrating the full set of copies to one or more computing environments that differ from the original source. The migration destination can be a non-cloud data center, a corporate cloud, a different cloud computing service, a different cloud availability zone, a different cloud service account, and/or any combination thereof, without limitation. The data storage management system, including the various components it deploys, can operate in one or more computing environments, e.g., one or more cloud computing environments, in a non-cloud data center, and/or any combination thereof, without limitation.

The illustrative data storage management system comprises an enhanced storage manager that is responsible for the discovery and orchestration functions, and for invoking cloud-based components of the system, such as data agents, media agents, and/or storage resources. The storage manager is a computing resource that is generally responsible for managing storage operations throughout the data storage management system and executes on a non-virtualized computing device or on a virtual machine in cloud or non-cloud environments, without limitation. Cloud services conversions, if needed, are also orchestrated by the enhanced storage manager. An enhanced file system data agent handles access to/from configuration file(s) for native cloud applications. The system also generates and stores new data structures in reference to the application entity that represents the serverless application. These data structures, such as the application entity definition, the asset mapping, the preferences, and the associations and indexing information, are stored to one or more locations within the system where they can be readily accessed. Illustratively, these data structures are stored in a management database associated with the storage manager, but the invention is not so limited.

In sum, the illustrative data storage management system holistically protects serverless applications that operate in various distributed environments, including single-cloud, multi-zone, multi-cloud, non-cloud data center, and/or any combination thereof. The data storage management system treats each serverless application as a collective application entity whose assets are protected as a related group in a substantially synchronous manner to represent point-in-time views of the serverless application.

The illustrative data storage management system is configured to restore the sets of point-in-time copies of the application assets to the same cloud computing environment(s) where the original serverless application operates, thus restoring the serverless application to its point-in-time view. The illustrative data storage management is further configured to migrate the serverless application by migrating, and converting if need be, the sets of point-in-time copies of the application assets to computing environments that differ from the application's source, including single-cloud, multi-zone, multi-cloud, non-cloud data center, and/or any combination thereof. System components, such as media agents and data agents are invoked for retrieving the copies from secondary storage and restoring them to operational form in the destination computing environment(s).

In some embodiments, the data storage management system is further configured to recommend an improved and coordinated schedule for protecting the various assets that are associated with one or more serverless applications. For example, if a certain database is associated with a serverless application, it will be backed up according to the preferences for the application-entity. But it might also be protected according to preferences for the database entity itself, thus creating separate rounds of backup operations for the same database. The data storage management system will generate recommendations for a coordinated schedule that unifies the database backups and reduces redundancies. The coordinated schedule will be recommended to system administrators for approval. In alternative embodiments, the coordinated schedule will be automatically implemented when the system detects a redundancy between application-entity protection and individual asset protection, without limitation.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1A is a block diagram illustrating an exemplary information management system.

FIG. 1B is a detailed view of a primary storage device, a secondary storage device, and some examples of primary data and secondary copy data.

FIG. 1C is a block diagram of an exemplary information management system including a storage manager, one or more data agents, and one or more media agents.

FIG. 1D is a block diagram illustrating a scalable information management system.

FIG. 1E illustrates certain secondary copy operations according to an exemplary storage policy.

FIGS. 1F-1H are block diagrams illustrating suitable data structures that may be employed by the information management system.

FIG. 2A illustrates a system and technique for synchronizing primary data to a destination such as a failover site using secondary copy data.

FIG. 2B illustrates an information management system architecture incorporating use of a network file system (NFS) protocol for communicating between the primary and secondary storage subsystems.

FIG. 2C is a block diagram of an example of a highly scalable managed data pool architecture.

FIG. 3A is a block diagram illustrating some salient portions of a system **300** for holistically protecting serverless applications in a cloud computing environment, according to an illustrative embodiment of the present invention.

FIG. 3B is a block diagram illustrating some salient portions of system **300** for holistically protecting serverless applications from a cloud computing environment, wherein system **300** operates outside the cloud, according to an illustrative embodiment of the present invention.

FIG. 3C is a block diagram illustrating some salient portions of system **300** for holistically protecting serverless applications from a plurality of cloud computing environments, wherein system **300** operates outside the cloud, according to an illustrative embodiment of the present invention.

FIG. 3D is a block diagram illustrating some salient portions of system **300** for holistically protecting a serverless application from a first cloud computing environment and migrating the serverless application to a second cloud computing environment, according to an illustrative embodiment of the present invention.

FIG. **4** is a block diagram illustrating some salient assets of a serverless application that is protected and restored by system **300**, according to an illustrative embodiment of the present invention.

FIG. **5**A is a block diagram illustrating some salient portions of system **300** protecting a serverless application comprising a plurality of assets, according to an illustrative embodiment of the present invention.

FIG. **5**B is a block diagram illustrating some salient portions of system **300**, including storage manager **540** and management database **546**, according to an illustrative embodiment of the present invention.

FIG. **6** depicts some salient operations of a method **600** according to an illustrative embodiment of the present invention.

FIG. **6** depicts some salient operations of block **602** in method **600**.

FIG. **7**A depicts some salient operations of a method **600** according to an illustrative embodiment of the present invention.

FIG. **7**B depicts some additional salient operations of block **602** in method **600**.

FIG. **8**A depicts some salient operations of block **606** in method **600**.

FIG. **8**B depicts some additional salient operations of block **606** in method **600**.

FIG. **9** depicts some salient operations of block **808** in method **600**.

FIG. **10** depicts some salient operations of block **610** in method **600**.

FIG. **11** depicts an illustrative order of operations.

FIG. **12** depicts an illustrative list (catalogue, inventory) of cloud computing environment assets of an application configured in a cloud account.

FIG. **13** depicts an illustrative serverless distributed application service definition in a cloud computing environment.

FIG. **14** depicts an illustrative service definition of a serverless application that is distributed across multiple cloud computing environments.

DETAILED DESCRIPTION

Detailed descriptions and examples of systems and methods according to one or more illustrative embodiments of the present invention may be found in the section entitled HOLISTICALLY PROTECTING SERVERLESS APPLICATIONS DISTRIBUTED ACROSS ONE OR MORE CLOUD COMPUTING ENVIRONMENTS, as well as in the section entitled Example Embodiments, and also in FIGS. **3A-14** herein. Furthermore, components and functionality for protecting serverless applications distributed across one or more cloud computing environments may be configured and/or incorporated into information management systems such as those described herein in FIGS. **1**A-**1**H and **2**A-**2**C.

Various embodiments described herein are intimately tied to, enabled by, and would not exist except for, computer technology. For example, discovering serverless application assets, orchestrating storage management operations therefor, and/or restoring/migrating serverless applications from sets if point-in-time copies as described herein in reference to various embodiments cannot reasonably be performed by humans alone, without the computer technology upon which they are implemented.

Information Management System Overview

With the increasing importance of protecting and leveraging data, organizations simply cannot risk losing critical data. Moreover, runaway data growth and other modern realities make protecting and managing data increasingly difficult. There is therefore a need for efficient, powerful, and user-friendly solutions for protecting and managing data and for smart and efficient management of data storage. Depending on the size of the organization, there may be many data production sources which are under the purview of tens, hundreds, or even thousands of individuals. In the past, individuals were sometimes responsible for managing and protecting their own data, and a patchwork of hardware and software point solutions may have been used in any given organization. These solutions were often provided by different vendors and had limited or no interoperability. Certain embodiments described herein address these and other shortcomings of prior approaches by implementing scalable, unified, organization-wide information management, including data storage management.

FIG. **1**A shows one such information management system **100** (or "system **100**"), which generally includes combinations of hardware and software configured to protect and manage data and metadata that are generated and used by computing devices in system **100**. System **100** may be referred to in some embodiments as a "storage management system" or a "data storage management system." System

100 performs information management operations, some of which may be referred to as "storage operations" or "data storage operations," to protect and manage the data residing in and/or managed by system **100**. The organization that employs system **100** may be a corporation or other business entity, non-profit organization, educational institution, household, governmental agency, or the like.

Generally, the systems and associated components described herein may be compatible with and/or provide some or all of the functionality of the systems and corresponding components described in one or more of the following U.S. patents/publications and patent applications assigned to Commvault Systems, Inc., each of which is hereby incorporated by reference in its entirety herein:

- U.S. Pat. No. 7,035,880, entitled "Modular Backup and Retrieval System Used in Conjunction With a Storage Area Network";
- U.S. Pat. No. 7,107,298, entitled "System And Method For Archiving Objects In An Information Store";
- U.S. Pat. No. 7,246,207, entitled "System and Method for Dynamically Performing Storage Operations in a Computer Network";
- U.S. Pat. No. 7,315,923, entitled "System And Method For Combining Data Streams In Pipelined Storage Operations In A Storage Network";
- U.S. Pat. No. 7,343,453, entitled "Hierarchical Systems and Methods for Providing a Unified View of Storage Information";
- U.S. Pat. No. 7,395,282, entitled "Hierarchical Backup and Retrieval System";
- U.S. Pat. No. 7,529,782, entitled "System and Methods for Performing a Snapshot and for Restoring Data";
- U.S. Pat. No. 7,617,262, entitled "System and Methods for Monitoring Application Data in a Data Replication System";
- U.S. Pat. No. 7,734,669, entitled "Managing Copies Of Data";
- U.S. Pat. No. 7,747,579, entitled "Metabase for Facilitating Data Classification";
- U.S. Pat. No. 8,156,086, entitled "Systems And Methods For Stored Data Verification";
- U.S. Pat. No. 8,170,995, entitled "Method and System for Offline Indexing of Content and Classifying Stored Data";
- U.S. Pat. No. 8,230,195, entitled "System And Method For Performing Auxiliary Storage Operations";
- U.S. Pat. No. 8,285,681, entitled "Data Object Store and Server for a Cloud Storage Environment, Including Data Deduplication and Data Management Across Multiple Cloud Storage Sites";
- U.S. Pat. No. 8,307,177, entitled "Systems And Methods For Management Of Virtualization Data";
- U.S. Pat. No. 8,364,652, entitled "Content-Aligned, Block-Based Deduplication";
- U.S. Pat. No. 8,578,120, entitled "Block-Level Single Instancing";
- U.S. Pat. No. 8,954,446, entitled "Client-Side Repository in a Networked Deduplicated Storage System";
- U.S. Pat. No. 9,020,900, entitled "Distributed Deduplicated Storage System";
- U.S. Pat. No. 9,098,495, entitled "Application-Aware and Remote Single Instance Data Management";
- U.S. Pat. No. 9,239,687, entitled "Systems and Methods for Retaining and Using Data Block Signatures in Data Protection Operations";
- U.S. Pat. No. 9,633,033, entitled "High Availability Distributed Deduplicated Storage System";
- U.S. Pat. Pub. No. 2006/0224846, entitled "System and Method to Support Single Instance Storage Operations";
- U.S. Pat. Pub. No. 2016-0350391, entitled "Replication Using Deduplicated Secondary Copy Data";
- U.S. Pat. Pub. No. 2017-0168903 A1, entitled "Live Synchronization and Management of Virtual Machines across Computing and Virtualization Platforms and Using Live Synchronization to Support Disaster Recovery";
- U.S. Pat. Pub. No. 2017-0185488 A1, entitled "Application-Level Live Synchronization Across Computing Platforms Including Synchronizing Co-Resident Applications To Disparate Standby Destinations And Selectively Synchronizing Some Applications And Not Others";
- U.S. Pat. Pub. No. 2017-0192866 A1, entitled "System For Redirecting Requests After A Secondary Storage Computing Device Failure";
- U.S. Pat. Pub. No. 2017-0235647 A1, entitled "Data Protection Operations Based on Network Path Information";

and

- U.S. Pat. Pub. No. 2017-0242871 A1, entitled "Data Restoration Operations Based on Network Path Information".

System **100** includes computing devices and computing technologies. For instance, system **100** can include one or more client computing devices **102** and secondary storage computing devices **106**, as well as storage manager **140** or a host computing device for it. Computing devices can include, without limitation, one or more: workstations, personal computers, desktop computers, or other types of generally fixed computing systems such as mainframe computers, servers, and minicomputers. Other computing devices can include mobile or portable computing devices, such as one or more laptops, tablet computers, personal data assistants, mobile phones (such as smartphones), and other mobile or portable computing devices such as embedded computers, set top boxes, vehicle-mounted devices, wearable computers, etc. Servers can include mail servers, file servers, database servers, virtual machine servers, and web servers. Any given computing device comprises one or more processors (e.g., CPU and/or single-core or multi-core processors), as well as corresponding non-transitory computer memory (e.g., random-access memory (RAM)) for storing computer programs which are to be executed by the one or more processors. Other computer memory for mass storage of data may be packaged/configured with the computing device (e.g., an internal hard disk) and/or may be external and accessible by the computing device (e.g., network-attached storage, a storage array, etc.). In some cases, a computing device includes cloud computing resources, which may be implemented as virtual machines. For instance, one or more virtual machines may be provided to the organization by a third-party cloud service vendor.

In some embodiments, computing devices can include one or more virtual machine(s) running on a physical host computing device (or "host machine") operated by the organization. As one example, the organization may use one virtual machine as a database server and another virtual machine as a mail server, both virtual machines operating on the same host machine. A Virtual machine ("VM") is a software implementation of a computer that does not physically exist and is instead instantiated in an operating system of a physical computer (or host machine) to enable applications to execute within the VM's environment, i.e., a VM emulates a physical computer. AVM includes an operating system and associated virtual resources, such as computer memory and processor(s). A hypervisor operates between the VM and the hardware of the physical host machine and is generally responsible for creating and running the VMs. Hypervisors are also known in the art as virtual machine monitors or a virtual machine managers or "VMMs", and may be implemented in software, firmware, and/or specialized hardware installed on the host machine. Examples of hypervisors include ESX Server, by VMware, Inc. of Palo Alto, Calif.; Microsoft Virtual Server and Microsoft Windows Server Hyper-V, both by Microsoft Corporation of Redmond, Wash.; Sun xVM by Oracle America Inc. of Santa Clara, Calif.; and Xen by Citrix Systems, Santa Clara, Calif. The hypervisor provides resources to each virtual operating system such as a virtual processor, virtual memory, a virtual network device, and a virtual disk. Each virtual machine has one or more associated virtual disks. The hypervisor typically stores the data of virtual disks in files on the file system of the physical host machine, called virtual machine disk files ("VMDK" in VMware lingo) or virtual hard disk image files (in Microsoft lingo). For example, VMware's ESX Server provides the Virtual Machine File System (VMFS) for the storage of virtual machine disk files. A virtual machine reads data from and writes data to its virtual disk much the way that a physical machine reads data from and writes data to a physical disk. Examples of techniques for implementing information management in a cloud computing environment are described in U.S. Pat. No. 8,285,681. Examples of techniques for implementing information management in a virtualized computing environment are described in U.S. Pat. No. 8,307,177.

Information management system **100** can also include electronic data storage devices, generally used for mass storage of data, including, e.g., primary storage devices **104** and secondary storage devices **108**. Storage devices can generally be of any suitable type including, without limitation, disk drives, storage arrays (e.g., storage-area network (SAN) and/or network-attached storage (NAS) technology), semiconductor memory (e.g., solid state storage devices), network attached storage (NAS) devices, tape libraries, or other magnetic, non-tape storage devices, optical media storage devices, combinations of the same, etc. In some embodiments, storage devices form part of a distributed file system. In some cases, storage devices are provided in a cloud storage environment (e.g., a private cloud or one operated by a third-party vendor), whether for primary data or secondary copies or both.

Depending on context, the term "information management system" can refer to generally all of the illustrated hardware and software components in FIG. **1**C, or the term may refer to only a subset of the illustrated components. For instance, in some cases, system **100** generally refers to a combination of specialized components used to protect, move, manage, manipulate, analyze, and/or process data and metadata generated by client computing devices **102**. However, system **100** in some cases does not include the underlying components that generate and/or store primary data **112**, such as the client computing devices **102** themselves, and the primary storage devices **104**. Likewise, secondary storage devices **108** (e.g., a third-party provided cloud storage environment) may not be part of system **100**. As an example, "information management system" or "storage management system" may sometimes refer to one or more of the following components, which will be described in further detail below: storage manager, data agent, and media agent.

One or more client computing devices **102** may be part of system **100**, each client computing device **102** having an operating system and at least one application **110** and one or more accompanying data agents executing thereon; and associated with one or more primary storage devices **104** storing primary data **112**. Client computing device(s) **102** and primary storage devices **104** may generally be referred to in some cases as primary storage subsystem **117**.

Client Computing Devices, Clients, and Subclients

Typically, a variety of sources in an organization produce data to be protected and managed. As just one illustrative example, in a corporate environment such data sources can be employee workstations and company servers such as a mail server, a web server, a database server, a transaction server, or the like. In system **100**, data generation sources include one or more client computing devices **102**. A computing device that has a data agent **142** installed and operating on it is generally referred to as a "client computing device" **102**, and may include any type of computing device, without limitation. A client computing device **102** may be associated with one or more users and/or user accounts.

A "client" is a logical component of information management system **100**, which may represent a logical grouping of one or more data agents installed on a client computing device **102**. Storage manager **140** recognizes a client as a component of system **100**, and in some embodiments, may automatically create a client component the first time a data agent **142** is installed on a client computing device **102**. Because data generated by executable component(s) **110** is tracked by the associated data agent **142** so that it may be properly protected in system **100**, a client may be said to generate data and to store the generated data to primary storage, such as primary storage device **104**. However, the terms "client" and "client computing device" as used herein do not imply that a client computing device **102** is necessarily configured in the client/server sense relative to another computing device such as a mail server, or that a client computing device **102** cannot be a server in its own right. As just a few examples, a client computing device **102** can be and/or include mail servers, file servers, database servers, virtual machine servers, and/or web servers.

Each client computing device **102** may have application(s) **110** executing thereon which generate and manipulate the data that is to be protected from loss and managed in system **100**. Applications **110** generally facilitate the operations of an organization, and can include, without limitation, mail server applications (e.g., Microsoft Exchange Server), file system applications, mail client applications (e.g., Microsoft Exchange Client), database applications or database management systems (e.g., SQL, Oracle, SAP, Lotus Notes Database), word processing applications (e.g., Microsoft Word), spreadsheet applications, financial applications, presentation applications, graphics and/or video applications, browser applications, mobile applications, entertainment applications, and so on. Each application **110** may be accompanied by an application-specific data agent **142**, though not all data agents **142** are application-specific or associated with only application. A file manager application, e.g., Microsoft Windows Explorer, may be considered an application **110** and may be accompanied by its own data agent **142**. Client computing devices **102** can have at least one operating system (e.g., Microsoft Windows, Mac OS X, iOS, IBM z/OS, Linux, other Unix-based operating systems, etc.) installed thereon, which may support or host one or more file systems and other applications **110**. In some embodiments, a virtual machine that executes on a host client computing device **102** may be considered an application **110** and may be accompanied by a specific data agent **142** (e.g., virtual server data agent).

Client computing devices **102** and other components in system **100** can be connected to one another via one or more electronic communication pathways **114**. For example, a first communication pathway **114** may communicatively couple client computing device **102** and secondary storage computing device **106**; a second communication pathway **114** may communicatively couple storage manager **140** and client computing device **102**; and a third communication pathway **114** may communicatively couple storage manager **140** and secondary storage computing device **106**, etc. (see, e.g., FIG. **1**A and FIG. **1**C). A communication pathway **114** can include one or more networks or other connection types including one or more of the following, without limitation: the Internet, a wide area network (WAN), a local area network (LAN), a Storage Area Network (SAN), a Fibre Channel (FC) connection, a Small Computer System Interface (SCSI) connection, a virtual private network (VPN), a token ring or TCP/IP based network, an intranet network, a point-to-point link, a cellular network, a wireless data transmission system, a two-way cable system, an interactive kiosk network, a satellite network, a broadband network, a baseband network, a neural network, a mesh network, an ad hoc network, other appropriate computer or telecommunications networks, combinations of the same or the like. Communication pathways **114** in some cases may also include application programming interfaces (APIs) including, e.g., cloud service provider APIs, virtual machine management APIs, and hosted service provider APIs. The underlying infrastructure of communication pathways **114** may be wired and/or wireless, analog and/or digital, or any combination thereof; and the facilities used may be private, public, third-party provided, or any combination thereof, without limitation.

A "subclient" is a logical grouping of all or part of a client's primary data **112**. In general, a subclient may be defined according to how the subclient data is to be protected as a unit in system **100**. For example, a subclient may be associated with a certain storage policy. A given client may thus comprise several subclients, each subclient associated with a different storage policy. For example, some files may form a first subclient that requires compression and deduplication and is associated with a first storage policy. Other files of the client may form a second subclient that requires a different retention schedule as well as encryption, and may be associated with a different, second storage policy. As a result, though the primary data may be generated by the same application **110** and may belong to one given client, portions of the data may be assigned to different subclients for distinct treatment by system **100**. More detail on subclients is given in regard to storage policies below.

Primary Data and Exemplary Primary Storage Devices

Primary data **112** is generally production data or "live" data generated by the operating system and/or applications **110** executing on client computing device **102**. Primary data **112** is generally stored on primary storage device(s) **104** and is organized via a file system operating on the client computing device **102**. Thus, client computing device(s) **102** and corresponding applications **110** may create, access, modify, write, delete, and otherwise use primary data **112**. Primary data **112** is generally in the native format of the source application **110**. Primary data **112** is an initial or first stored body of data generated by the source application **110**. Primary data **112** in some cases is created substantially directly from data generated by the corresponding source application **110**. It can be useful in performing certain tasks to organize primary data **112** into units of different granularities. In general, primary data **112** can include files, directories, file system volumes, data blocks, extents, or any other hierarchies or organizations of data objects. As used herein, a "data object" can refer to (i) any file that is currently addressable by a file system or that was previously addressable by the file system (e.g., an archive file), and/or to (ii) a subset of such a file (e.g., a data block, an extent, etc.). Primary data **112** may include structured data (e.g., database files), unstructured data (e.g., documents), and/or semi-structured data. See, e.g., FIG. **1**B.

It can also be useful in performing certain functions of system **100** to access and modify metadata within primary data **112**. Metadata generally includes information about data objects and/or characteristics associated with the data objects. For simplicity herein, it is to be understood that, unless expressly stated otherwise, any reference to primary data **112** generally also includes its associated metadata, but references to metadata generally do not include the primary data. Metadata can include, without limitation, one or more of the following: the data owner (e.g., the client or user that generates the data), the last modified time (e.g., the time of the most recent modification of the data object), a data object name (e.g., a file name), a data object size (e.g., a number of bytes of data), information about the content (e.g., an indication as to the existence of a particular search term), user-supplied tags, to/from information for email (e.g., an email sender, recipient, etc.), creation date, file type (e.g., format or application type), last accessed time, application type (e.g., type of application that generated the data object), location/network (e.g., a current, past or future location of the data object and network pathways to/from the data object), geographic location (e.g., GPS coordinates), frequency of change (e.g., a period in which the data object is modified), business unit (e.g., a group or department that generates, manages or is otherwise associated with the data object), aging information (e.g., a schedule, such as a time period, in which the data object is migrated to secondary or long term storage), boot sectors, partition layouts, file location within a file folder directory structure, user permissions, owners, groups, access control lists (ACLs), system metadata (e.g., registry information), combinations of the same or other similar information related to the data object. In addition to metadata generated by or related to file systems and operating systems, some applications **110** and/or other components of system **100** maintain indices of metadata for data objects, e.g., metadata associated with individual email messages. The use of metadata to perform classification and other functions is described in greater detail below.

Primary storage devices **104** storing primary data **112** may be relatively fast and/or expensive technology (e.g., flash storage, a disk drive, a hard-disk storage array, solid state memory, etc.), typically to support high-performance live production environments. Primary data **112** may be highly changeable and/or may be intended for relatively short term retention (e.g., hours, days, or weeks). According to some embodiments, client computing device **102** can access primary data **112** stored in primary storage device **104** by making conventional file system calls via the operating system. Each client computing device **102** is generally associated with and/or in communication with one or more primary storage devices **104** storing corresponding primary data **112**. A client computing device **102** is said to be associated with or in communication with a particular primary storage device **104** if it is capable of one or more of: routing and/or storing data (e.g., primary data **112**) to the primary storage device **104**, coordinating the routing and/or storing of data to the primary storage device **104**, retrieving data from the primary storage device **104**, coordinating the retrieval of data from the primary storage device **104**, and modifying and/or deleting data in the primary storage device **104**. Thus, a client computing device **102** may be said to access data stored in an associated storage device **104**.

Primary storage device **104** may be dedicated or shared. In some cases, each primary storage device **104** is dedicated to an associated client computing device **102**, e.g., a local disk drive. In other cases, one or more primary storage devices **104** can be shared by multiple client computing devices **102**, e.g., via a local network, in a cloud storage implementation, etc. As one example, primary storage device **104** can be a storage array shared by a group of client computing devices **102**, such as EMC Clariion, EMC Symmetrix, EMC Celerra, Dell EqualLogic, IBM XIV, NetApp FAS, HP EVA, and HP 3PAR.

System **100** may also include hosted services (not shown), which may be hosted in some cases by an entity other than the organization that employs the other components of system **100**. For instance, the hosted services may be provided by online service providers. Such service providers can provide social networking services, hosted email services, or hosted productivity applications or other hosted applications such as software-as-a-service (SaaS), platform-as-a-service (PaaS), application service providers (ASPs), cloud services, or other mechanisms for delivering functionality via a network. As it services users, each hosted service may generate additional data and metadata, which may be managed by system **100**, e.g., as primary data **112**. In some cases, the hosted services may be accessed using one of the applications **110**. As an example, a hosted mail service may be accessed via browser running on a client computing device **102**.

Secondary Copies and Exemplary Secondary Storage Devices

Primary data **112** stored on primary storage devices **104** may be compromised in some cases, such as when an employee deliberately or accidentally deletes or overwrites primary data **112**. Or primary storage devices **104** can be damaged, lost, or otherwise corrupted. For recovery and/or regulatory compliance purposes, it is therefore useful to generate and maintain copies of primary data **112**. Accordingly, system **100** includes one or more secondary storage computing devices **106** and one or more secondary storage devices **108** configured to create and store one or more secondary copies **116** of primary data **112** including its associated metadata. The secondary storage computing devices **106** and the secondary storage devices **108** may be referred to as secondary storage subsystem **118**.

Secondary copies **116** can help in search and analysis efforts and meet other information management goals as well, such as: restoring data and/or metadata if an original version is lost (e.g., by deletion, corruption, or disaster); allowing point-in-time recovery; complying with regulatory data retention and electronic discovery (e-discovery) requirements; reducing utilized storage capacity in the production system and/or in secondary storage; facilitating organization and search of data; improving user access to data files across multiple computing devices and/or hosted services; and implementing data retention and pruning policies.

A secondary copy **116** can comprise a separate stored copy of data that is derived from one or more earlier-created stored copies (e.g., derived from primary data **112** or from another secondary copy **116**). Secondary copies **116** can include point-in-time data and may be intended for relatively long-term retention before some or all of the data is moved to other storage or discarded. In some cases, a secondary copy **116** may be in a different storage device than other previously stored copies; and/or may be remote from other previously stored copies. Secondary copies **116** can be stored in the same storage device as primary data **112**. For example, a disk array capable of performing hardware snapshots stores primary data **112** and creates and stores hardware snapshots of the primary data **112** as secondary copies **116**. Secondary copies **116** may be stored in relatively slow and/or lower cost storage (e.g., magnetic tape). A secondary copy **116** may be stored in a backup or archive format, or in some other format different from the native source application format or other format of primary data **112**.

Secondary storage computing devices **106** may index secondary copies **116** (e.g., using a media agent **144**), enabling users to browse and restore at a later time and further enabling the lifecycle management of the indexed data. After creation of a secondary copy **116** that represents certain primary data **112**, a pointer or other location indicia (e.g., a stub) may be placed in primary data **112**, or be otherwise associated with primary data **112**, to indicate the current location of a particular secondary copy **116**. Since an instance of a data object or metadata in primary data **112** may change over time as it is modified by application **110** (or hosted service or the operating system), system **100** may create and manage multiple secondary copies **116** of a particular data object or metadata, each copy representing the state of the data object in primary data **112** at a particular point in time. Moreover, since an instance of a data object in primary data **112** may eventually be deleted from primary storage device **104** and the file system, system **100** may continue to manage point-in-time representations of that data object, even though the instance in primary data **112** no longer exists. For virtual machines, the operating system and other applications **110** of client computing device(s) **102** may execute within or under the management of virtualization software (e.g., a VMM), and the primary storage device(s) **104** may comprise a virtual disk created on a physical storage device. System **100** may create secondary copies **116** of the files or other data objects in a virtual disk file and/or secondary copies **116** of the entire virtual disk file itself (e.g., of an entire .vmdk file).

Secondary copies **116** are distinguishable from corresponding primary data **112**. First, secondary copies **116** can be stored in a different format from primary data **112** (e.g., backup, archive, or another non-native format). For this or other reasons, secondary copies **116** may not be directly usable by applications **110** or client computing device **102** (e.g., via standard system calls or otherwise) without modification, processing, or other intervention by system **100** which may be referred to as "restore" operations. Secondary copies **116** may have been processed by data agent **142** and/or media agent **144** in the course of being created (e.g., compression, deduplication, encryption, integrity markers, indexing, formatting, application-aware metadata, etc.), and thus secondary copy **116** may represent source primary data **112** without necessarily being exactly identical to the source.

Second, secondary copies **116** may be stored on a secondary storage device **108** that is inaccessible to application **110** running on client computing device **102** and/or hosted service. Some secondary copies **116** may be "offline copies," in that they are not readily available (e.g., not mounted to tape or disk). Offline copies can include copies of data that system **100** can access without human intervention (e.g., tapes within an automated tape library, but not yet mounted in a drive), and copies that the system **100** can access only with some human intervention (e.g., tapes located at an offsite storage site).

Using Intermediate Devices for Creating Secondary Copies—Secondary Storage Computing Devices

Creating secondary copies can be challenging when hundreds or thousands of client computing devices **102** continually generate large volumes of primary data **112** to be protected. Also, there can be significant overhead involved in the creation of secondary copies **116**. Moreover, specialized programmed intelligence and/or hardware capability is generally needed for accessing and interacting with secondary storage devices **108**. Client computing devices **102** may interact directly with a secondary storage device **108** to create secondary copies **116**, but in view of the factors described above, this approach can negatively impact the ability of client computing device **102** to serve/service application **110** and produce primary data **112**. Further, any given client computing device **102** may not be optimized for interaction with certain secondary storage devices **108**.

Thus, system **100** may include one or more software and/or hardware components which generally act as intermediaries between client computing devices **102** (that generate primary data **112**) and secondary storage devices **108** (that store secondary copies **116**). In addition to off-loading certain responsibilities from client computing devices **102**, these intermediate components provide other benefits. For instance, as discussed further below with respect to FIG. **1**D, distributing some of the work involved in creating secondary copies **116** can enhance scalability and improve system performance. For instance, using specialized secondary storage computing devices **106** and media agents **144** for interfacing with secondary storage devices **108** and/or for performing certain data processing operations can greatly improve the speed with which system **100** performs information management operations and can also improve the capacity of the system to handle large numbers of such operations, while reducing the computational load on the production environment of client computing devices **102**. The intermediate components can include one or more secondary storage computing devices **106** as shown in FIG. **1**A and/or one or more media agents **144**. Media agents are discussed further below (e.g., with respect to FIGS. **1**C-**1**E). These special-purpose components of system **100** comprise specialized programmed intelligence and/or hardware capability for writing to, reading from, instructing, communicating with, or otherwise interacting with secondary storage devices **108**.

Secondary storage computing device(s) **106** can comprise any of the computing devices described above, without limitation. In some cases, secondary storage computing device(s) **106** also include specialized hardware componentry and/or software intelligence (e.g., specialized interfaces) for interacting with certain secondary storage device(s) **108** with which they may be specially associated.

To create a secondary copy **116** involving the copying of data from primary storage subsystem **117** to secondary storage subsystem **118**, client computing device **102** may communicate the primary data **112** to be copied (or a processed version thereof generated by a data agent **142**) to the designated secondary storage computing device **106**, via a communication pathway **114**. Secondary storage computing device **106** in turn may further process and convey the data or a processed version thereof to secondary storage device **108**. One or more secondary copies **116** may be created from existing secondary copies **116**, such as in the case of an auxiliary copy operation, described further below.

Exemplary Primary Data and an Exemplary Secondary Copy

FIG. **16** is a detailed view of some specific examples of primary data stored on primary storage device(s) **104** and secondary copy data stored on secondary storage device(s) **108**, with other components of the system removed for the purposes of illustration. Stored on primary storage device(s) **104** are primary data **112** objects including word processing documents **119**A-B, spreadsheets **120**, presentation documents **122**, video files **124**, image files **126**, email mailboxes **128** (and corresponding email messages **129**A-C), HTML/XML or other types of markup language files **130**, databases **132** and corresponding tables or other data structures **133**A-**133**C. Some or all primary data **112** objects are associated with corresponding metadata (e.g., "Meta1-11"), which may include file system metadata and/or application-specific metadata. Stored on the secondary storage device(s) **108** are secondary copy **116** data objects **134**A-C which may include copies of or may otherwise represent corresponding primary data **112**.

Secondary copy data objects **134**A-C can individually represent more than one primary data object. For example, secondary copy data object **134**A represents three separate primary data objects **133**C, **122**, and **129**C (represented as **133**C', **122'**, and **129**C', respectively, and accompanied by corresponding metadata Meta11, Meta3, and Meta8, respectively). Moreover, as indicated by the prime mark ('), secondary storage computing devices **106** or other components in secondary storage subsystem **118** may process the data received from primary storage subsystem **117** and store a secondary copy including a transformed and/or supplemented representation of a primary data object and/or metadata that is different from the original format, e.g., in a compressed, encrypted, deduplicated, or other modified format. For instance, secondary storage computing devices **106** can generate new metadata or other information based on said processing and store the newly generated information along with the secondary copies. Secondary copy data object **1346** represents primary data objects **120**, **1336**, and **119**A as **120'**, **1336'**, and **119**A', respectively, accompanied by corresponding metadata Meta2, Meta10, and Meta1, respectively. Also, secondary copy data object **134**C represents primary data objects **133**A, **1196**, and **129**A as **133**A', **1196'**, and **129**A', respectively, accompanied by corresponding metadata Meta9, Meta5, and Meta6, respectively.

Exemplary Information Management System Architecture

System **100** can incorporate a variety of different hardware and software components, which can in turn be organized with respect to one another in many different configurations, depending on the embodiment. There are critical design choices involved in specifying the functional responsibilities of the components and the role of each component in system **100**. Such design choices can impact how system **100** performs and adapts to data growth and other changing circumstances. FIG. **1**C shows a system **100** designed according to these considerations and includes: storage manager **140**, one or more data agents **142** executing on client computing device(s) **102** and configured to process primary data **112**, and one or more media agents **144** executing on one or more secondary storage computing devices **106** for performing tasks involving secondary storage devices **108**.

Storage Manager

Storage manager **140** is a centralized storage and/or information manager that is configured to perform certain control functions and also to store certain critical information about system **100**—hence storage manager **140** is said to manage system **100**. As noted, the number of components in system **100** and the amount of data under management can be large. Managing the components and data is therefore a significant task, which can grow unpredictably as the number of components and data scale to meet the needs of the organization. For these and other reasons, according to certain embodiments, responsibility for controlling system **100**, or at least a significant portion of that responsibility, is allocated to storage manager **140**. Storage manager **140** can be adapted independently according to changing circumstances, without having to replace or re-design the remainder of the system. Moreover, a computing device for hosting and/or operating as storage manager **140** can be selected to best suit the functions and networking needs of storage manager **140**. These and other advantages are described in further detail below and with respect to FIG. **1**D.

Storage manager **140** may be a software module or other application hosted by a suitable computing device. In some embodiments, storage manager **140** is itself a computing device that performs the functions described herein. Storage manager **140** comprises or operates in conjunction with one or more associated data structures such as a dedicated database (e.g., management database **146**), depending on the configuration. The storage manager **140** generally initiates, performs, coordinates, and/or controls storage and other information management operations performed by system **100**, e.g., to protect and control primary data **112** and secondary copies **116**. In general, storage manager **140** is said to manage system **100**, which includes communicating with, instructing, and controlling in some circumstances components such as data agents **142** and media agents **144**, etc.

As shown by the dashed arrowed lines **114** in FIG. **1**C, storage manager **140** may communicate with, instruct, and/or control some or all elements of system **100**, such as data agents **142** and media agents **144**. In this manner, storage manager **140** manages the operation of various hardware and software components in system **100**. In certain embodiments, control information originates from storage manager **140** and status as well as index reporting is transmitted to storage manager **140** by the managed components, whereas payload data and metadata are generally communicated between data agents **142** and media agents **144** (or otherwise between client computing device(s) **102** and secondary storage computing device(s) **106**), e.g., at the direction of and under the management of storage manager **140**. Control information can generally include parameters and instructions for carrying out information management operations, such as, without limitation, instructions to perform a task associated with an operation, timing information specifying when to initiate a task, data path information specifying what components to communicate with or access in carrying out an operation, and the like. In other embodiments, some information management operations are controlled or initiated by other components of system **100** (e.g., by media agents **144** or data agents **142**), instead of or in combination with storage manager **140**.

According to certain embodiments, storage manager **140** provides one or more of the following functions:

- communicating with data agents **142** and media agents **144**, including transmitting instructions, messages, and/or queries, as well as receiving status reports, index information, messages, and/or queries, and responding to same;
- initiating execution of information management operations;
- initiating restore and recovery operations;
- managing secondary storage devices **108** and inventory/capacity of the same;
- allocating secondary storage devices **108** for secondary copy operations;
- reporting, searching, and/or classification of data in system **100**;
- monitoring completion of and status reporting related to information management operations and jobs;
- tracking movement of data within system **100**;
- tracking age information relating to secondary copies **116**, secondary storage devices **108**, comparing the age information against retention guidelines, and initiating data pruning when appropriate;
- tracking logical associations between components in system **100**;
- protecting metadata associated with system **100**, e.g., in management database **146**;
- implementing job management, schedule management, event management, alert management, reporting, job history maintenance, user security management, disaster recovery management, and/or user interfacing for system administrators and/or end users of system **100**;
- sending, searching, and/or viewing of log files; and
- implementing operations management functionality.

Storage manager **140** may maintain an associated database **146** (or "storage manager database **146**" or "management database **146**") of management-related data and information management policies **148**. Database **146** is stored in computer memory accessible by storage manager **140**. Database **146** may include a management index **150** (or "index **150**") or other data structure(s) that may store: logical associations between components of the system; user preferences and/or profiles (e.g., preferences regarding encryption, compression, or deduplication of primary data or secondary copies; preferences regarding the scheduling, type, or other aspects of secondary copy or other operations; mappings of particular information management users or user accounts to certain computing devices or other components, etc.; management tasks; media containerization; other useful data; and/or any combination thereof. For example, storage manager **140** may use index **150** to track logical associations between media agents **144** and secondary storage devices **108** and/or movement of data to/from secondary storage devices **108**. For instance, index **150** may store data associating a client computing device **102** with a particular media agent **144** and/or secondary storage device **108**, as specified in an information management policy **148**.

Administrators and others may configure and initiate certain information management operations on an individual basis. But while this may be acceptable for some recovery operations or other infrequent tasks, it is often not workable for implementing on-going organization-wide data protection and management. Thus, system **100** may utilize information management policies **148** for specifying and executing information management operations on an automated basis. Generally, an information management policy **148** can include a stored data structure or other information source that specifies parameters (e.g., criteria and rules) associated with storage management or other information management operations. Storage manager **140** can process an information management policy **148** and/or index **150** and, based on the results, identify an information management operation to perform, identify the appropriate components in system **100** to be involved in the operation (e.g., client computing devices **102** and corresponding data agents **142**, secondary storage computing devices **106** and corresponding media agents **144**, etc.), establish connections to those components and/or between those components, and/or instruct and control those components to carry out the operation. In this manner, system **100** can translate stored information into coordinated activity among the various computing devices in system **100**.

Management database **146** may maintain information management policies **148** and associated data, although information management policies **148** can be stored in computer memory at any appropriate location outside management database **146**. For instance, an information management policy **148** such as a storage policy may be stored as metadata in a media agent database **152** or in a secondary storage device **108** (e.g., as an archive copy) for use in restore or other information management operations, depending on the embodiment. Information management policies **148** are described further below. According to certain embodiments, management database **146** comprises a relational database (e.g., an SQL database) for tracking metadata, such as metadata associated with secondary copy operations (e.g., what client computing devices **102** and corresponding subclient data were protected and where the secondary copies are stored and which media agent **144** performed the storage operation(s)). This and other metadata may additionally be stored in other locations, such as at secondary storage computing device **106** or on the secondary storage device **108**, allowing data recovery without the use of storage manager **140** in some cases. Thus, management database **146** may comprise data needed to kick off secondary copy operations (e.g., storage policies, schedule policies, etc.), status and reporting information about completed jobs (e.g., status and error reports on yesterday's backup jobs), and additional information sufficient to enable restore and disaster recovery operations (e.g., media agent associations, location indexing, content indexing, etc.).

Storage manager **140** may include a jobs agent **156**, a user interface **158**, and a management agent **154**, all of which may be implemented as interconnected software modules or application programs. These are described further below.

Jobs agent **156** in some embodiments initiates, controls, and/or monitors the status of some or all information management operations previously performed, currently being performed, or scheduled to be performed by system **100**. A job is a logical grouping of information management operations such as daily storage operations scheduled for a certain set of subclients (e.g., generating incremental block-level backup copies **116** at a certain time every day for database files in a certain geographical location). Thus, jobs agent **156** may access information management policies **148** (e.g., in management database **146**) to determine when, where, and how to initiate/control jobs in system **100**.

Storage Manager User Interfaces

User interface **158** may include information processing and display software, such as a graphical user interface (GUI), an application program interface (API), and/or other interactive interface(s) through which users and system processes can retrieve information about the status of information management operations or issue instructions to storage manager **140** and other components. Via user interface **158**, users may issue instructions to the components in system **100** regarding performance of secondary copy and recovery operations. For example, a user may modify a schedule concerning the number of pending secondary copy operations. As another example, a user may employ the GUI to view the status of pending secondary copy jobs or to monitor the status of certain components in system **100** (e.g., the amount of capacity left in a storage device). Storage manager **140** may track information that permits it to select, designate, or otherwise identify content indices, deduplication databases, or similar databases or resources or data sets within its information management cell (or another cell) to be searched in response to certain queries. Such queries may be entered by the user by interacting with user interface **158**.

Various embodiments of information management system **100** may be configured and/or designed to generate user interface data usable for rendering the various interactive user interfaces described. The user interface data may be used by system **100** and/or by another system, device, and/or software program (for example, a browser program), to render the interactive user interfaces. The interactive user interfaces may be displayed on, for example, electronic displays (including, for example, touch-enabled displays), consoles, etc., whether direct-connected to storage manager **140** or communicatively coupled remotely, e.g., via an internet connection. The present disclosure describes various embodiments of interactive and dynamic user interfaces, some of which may be generated by user interface agent **158**, and which are the result of significant technological development. The user interfaces described herein may provide improved human-computer interactions, allowing for significant cognitive and ergonomic efficiencies and advantages over previous systems, including reduced mental workloads, improved decision-making, and the like. User interface **158** may operate in a single integrated view or console (not shown). The console may support a reporting capability for generating a variety of reports, which may be tailored to a particular aspect of information management.

User interfaces are not exclusive to storage manager **140** and in some embodiments a user may access information locally from a computing device component of system **100**. For example, some information pertaining to installed data agents **142** and associated data streams may be available from client computing device **102**. Likewise, some information pertaining to media agents **144** and associated data streams may be available from secondary storage computing device **106**.

Storage Manager Management Agent

Management agent **154** can provide storage manager **140** with the ability to communicate with other components within system **100** and/or with other information management cells via network protocols and application programming interfaces (APIs) including, e.g., HTTP, HTTPS, FTP, REST, virtualization software APIs, cloud service provider APIs, and hosted service provider APIs, without limitation. Management agent **154** also allows multiple information management cells to communicate with one another. For example, system **100** in some cases may be one information management cell in a network of multiple cells adjacent to one another or otherwise logically related, e.g., in a WAN or LAN. With this arrangement, the cells may communicate with one another through respective management agents **154**. Inter-cell communications and hierarchy is described in greater detail in e.g., U.S. Pat. No. 7,343,453.

Information Management Cell

An "information management cell" (or "storage operation cell" or "cell") may generally include a logical and/or physical grouping of a combination of hardware and software components associated with performing information management operations on electronic data, typically one storage manager **140** and at least one data agent **142** (executing on a client computing device **102**) and at least one media agent **144** (executing on a secondary storage computing device **106**). For instance, the components shown in FIG. **1C** may together form an information management cell. Thus, in some configurations, a system **100** may be referred to as an information management cell or a storage operation cell. A given cell may be identified by the identity of its storage manager **140**, which is generally responsible for managing the cell.

Multiple cells may be organized hierarchically, so that cells may inherit properties from hierarchically superior cells or be controlled by other cells in the hierarchy (automatically or otherwise). Alternatively, in some embodiments, cells may inherit or otherwise be associated with information management policies, preferences, information management operational parameters, or other properties or characteristics according to their relative position in a hierarchy of cells. Cells may also be organized hierarchically according to function, geography, architectural considerations, or other factors useful or desirable in performing information management operations. For example, a first cell may represent a geographic segment of an enterprise, such as a Chicago office, and a second cell may represent a different geographic segment, such as a New York City office. Other cells may represent departments within a particular office, e.g., human resources, finance, engineering, etc. Where delineated by function, a first cell may perform one or more first types of information management operations (e.g., one or more first types of secondary copies at a certain frequency), and a second cell may perform one or more second types of information management operations (e.g., one or more second types of secondary copies at a different frequency and under different retention rules). In general, the hierarchical information is maintained by one or more storage managers **140** that manage the respective cells (e.g., in corresponding management database(s) **146**).

Data Agents

A variety of different applications **110** can operate on a given client computing device **102**, including operating systems, file systems, database applications, e-mail applications, and virtual machines, just to name a few. And, as part of the process of creating and restoring secondary copies **116**, the client computing device **102** may be tasked with processing and preparing the primary data **112** generated by these various applications **110**. Moreover, the nature of the processing/preparation can differ across application types, e.g., due to inherent structural, state, and formatting differences among applications **110** and/or the operating system of client computing device **102**. Each data agent **142** is therefore advantageously configured in some embodiments to assist in the performance of information management operations based on the type of data that is being protected at a client-specific and/or application-specific level.

Data agent **142** is a component of information system **100** and is generally directed by storage manager **140** to participate in creating or restoring secondary copies **116**. Data agent **142** may be a software program (e.g., in the form of a set of executable binary files) that executes on the same client computing device **102** as the associated application **110** that data agent **142** is configured to protect. Data agent **142** is generally responsible for managing, initiating, or otherwise assisting in the performance of information management operations in reference to its associated application(s) **110** and corresponding primary data **112** which is generated/accessed by the particular application(s) **110**. For instance, data agent **142** may take part in copying, archiving, migrating, and/or replicating of certain primary data **112** stored in the primary storage device(s) **104**. Data agent **142** may receive control information from storage manager **140**, such as commands to transfer copies of data objects and/or metadata to one or more media agents **144**. Data agent **142** also may compress, deduplicate, and encrypt certain primary data **112**, as well as capture application-related metadata before transmitting the processed data to media agent **144**. Data agent **142** also may receive instructions from storage manager **140** to restore (or assist in restoring) a secondary copy **116** from secondary storage device **108** to primary storage **104**, such that the restored data may be properly accessed by application **110** in a suitable format as though it were primary data **112**.

Each data agent **142** may be specialized for a particular application **110**. For instance, different individual data agents **142** may be designed to handle Microsoft Exchange data, Lotus Notes data, Microsoft Windows file system data, Microsoft Active Directory Objects data, SQL Server data, SharePoint data, Oracle database data, SAP database data, virtual machines and/or associated data, and other types of data. A file system data agent, for example, may handle data files and/or other file system information. If a client computing device **102** has two or more types of data **112**, a specialized data agent **142** may be used for each data type. For example, to backup, migrate, and/or restore all of the data on a Microsoft Exchange server, the client computing device **102** may use: (1) a Microsoft Exchange Mailbox data agent **142** to back up the Exchange mailboxes; (2) a Microsoft Exchange Database data agent **142** to back up the Exchange databases; (3) a Microsoft Exchange Public Folder data agent **142** to back up the Exchange Public Folders; and (4) a Microsoft Windows File System data agent **142** to back up the file system of client computing device **102**. In this example, these specialized data agents **142** are treated as four separate data agents **142** even though they operate on the same client computing device **102**. Other examples may include archive management data agents such as a migration archiver or a compliance archiver, Quick Recovery® agents, and continuous data replication agents. Application-specific data agents **142** can provide improved performance as compared to generic agents. For instance, because application-specific data agents **142** may only handle data for a single software application, the design, operation, and performance of the data agent **142** can be streamlined. The data agent **142** may therefore execute faster and consume less persistent storage and/or operating memory than data agents designed to generically accommodate multiple different software applications **110**.

Each data agent **142** may be configured to access data and/or metadata stored in the primary storage device(s) **104** associated with data agent **142** and its host client computing device **102** and process the data appropriately. For example, during a secondary copy operation, data agent **142** may arrange or assemble the data and metadata into one or more files having a certain format (e.g., a particular backup or archive format) before transferring the file(s) to a media agent **144** or another component. The file(s) may include a list of files or other metadata. In some embodiments, a data agent **142** may be distributed between client computing device **102** and storage manager **140** (and any other intermediate components) or may be deployed from a remote location or its functions approximated by a remote process that performs some or all of the functions of data agent **142**. In addition, a data agent **142** may perform some functions provided by media agent **144**. Other embodiments may employ one or more generic data agents **142** that can handle and process data from two or more different applications **110**, or that can handle and process multiple data types, instead of or in addition to using specialized data agents **142**. For example, one generic data agent **142** may be used to back up, migrate and restore Microsoft Exchange Mailbox data and Microsoft Exchange Database data, while another generic data agent may handle Microsoft Exchange Public Folder data and Microsoft Windows File System data.

Media Agents

As noted, off-loading certain responsibilities from client computing devices **102** to intermediate components such as secondary storage computing device(s) **106** and corresponding media agent(s) **144** can provide a number of benefits including improved performance of client computing device **102**, faster and more reliable information management operations, and enhanced scalability. In one example which will be discussed further below, media agent **144** can act as a local cache of recently-copied data and/or metadata stored to secondary storage device(s) **108**, thus improving restore capabilities and performance for the cached data.

Media agent **144** is a component of system **100** and is generally directed by storage manager **140** in creating and restoring secondary copies **116**. Whereas storage manager **140** generally manages system **100** as a whole, media agent **144** provides a portal to certain secondary storage devices **108**, such as by having specialized features for communicating with and accessing certain associated secondary storage device **108**. Media agent **144** may be a software program (e.g., in the form of a set of executable binary files) that executes on a secondary storage computing device **106**. Media agent **144** generally manages, coordinates, and facilitates the transmission of data between a data agent **142** (executing on client computing device **102**) and secondary storage device(s) **108** associated with media agent **144**. For instance, other components in the system may interact with media agent **144** to gain access to data stored on associated secondary storage device(s) **108**, (e.g., to browse, read, write, modify, delete, or restore data). Moreover, media agents **144** can generate and store information relating to characteristics of the stored data and/or metadata, or can generate and store other types of information that generally provides insight into the contents of the secondary storage devices **108**—generally referred to as indexing of the stored secondary copies **116**. Each media agent **144** may operate on a dedicated secondary storage computing device **106**, while in other embodiments a plurality of media agents **144** may operate on the same secondary storage computing device **106**.

A media agent **144** may be associated with a particular secondary storage device **108** if that media agent **144** is capable of one or more of: routing and/or storing data to the particular secondary storage device **108**; coordinating the routing and/or storing of data to the particular secondary storage device **108**; retrieving data from the particular secondary storage device **108**; coordinating the retrieval of data from the particular secondary storage device **108**; and modifying and/or deleting data retrieved from the particular secondary storage device **108**. Media agent **144** in certain embodiments is physically separate from the associated secondary storage device **108**. For instance, a media agent **144** may operate on a secondary storage computing device **106** in a distinct housing, package, and/or location from the associated secondary storage device **108**. In one example, a media agent **144** operates on a first server computer and is in communication with a secondary storage device(s) **108** operating in a separate rack-mounted RAID-based system.

A media agent **144** associated with a particular secondary storage device **108** may instruct secondary storage device **108** to perform an information management task. For instance, a media agent **144** may instruct a tape library to use a robotic arm or other retrieval means to load or eject a certain storage media, and to subsequently archive, migrate, or retrieve data to or from that media, e.g., for the purpose of restoring data to a client computing device **102**. As another example, a secondary storage device **108** may include an array of hard disk drives or solid state drives organized in a RAID configuration, and media agent **144** may forward a logical unit number (LUN) and other appropriate information to the array, which uses the received information to execute the desired secondary copy operation. Media agent **144** may communicate with a secondary storage device **108** via a suitable communications link, such as a SCSI or Fibre Channel link.

Each media agent **144** may maintain an associated media agent database **152**. Media agent database **152** may be stored to a disk or other storage device (not shown) that is local to the secondary storage computing device **106** on which media agent **144** executes. In other cases, media agent database **152** is stored separately from the host secondary storage computing device **106**. Media agent database **152** can include, among other things, a media agent index **153** (see, e.g., FIG. **1C**). In some cases, media agent index **153** does not form a part of and is instead separate from media agent database **152**.

Media agent index **153** (or "index **153**") may be a data structure associated with the particular media agent **144** that includes information about the stored data associated with the particular media agent and which may be generated in the course of performing a secondary copy operation or a restore. Index **153** provides a fast and efficient mechanism for locating/browsing secondary copies **116** or other data stored in secondary storage devices **108** without having to access secondary storage device **108** to retrieve the information from there. For instance, for each secondary copy **116**, index **153** may include metadata such as a list of the data objects (e.g., files/subdirectories, database objects, mailbox objects, etc.), a logical path to the secondary copy **116** on the corresponding secondary storage device **108**, location information (e.g., offsets) indicating where the data objects are stored in the secondary storage device **108**, when the data objects were created or modified, etc. Thus, index **153** includes metadata associated with the secondary copies **116** that is readily available for use from media agent **144**. In some embodiments, some or all of the information in index **153** may instead or additionally be stored along with secondary copies **116** in secondary storage device **108**. In some embodiments, a secondary storage device **108** can include sufficient information to enable a "bare metal restore," where the operating system and/or software applications of a failed client computing device **102** or another target may be automatically restored without manually reinstalling individual software packages (including operating systems).

Because index **153** may operate as a cache, it can also be referred to as an "index cache." In such cases, information stored in index cache **153** typically comprises data that reflects certain particulars about relatively recent secondary copy operations. After some triggering event, such as after some time elapses or index cache **153** reaches a particular size, certain portions of index cache **153** may be copied or migrated to secondary storage device **108**, e.g., on a least-recently-used basis. This information may be retrieved and uploaded back into index cache **153** or otherwise restored to media agent **144** to facilitate retrieval of data from the secondary storage device(s) **108**. In some embodiments, the cached information may include format or containerization information related to archives or other files stored on storage device(s) **108**.

In some alternative embodiments media agent **144** generally acts as a coordinator or facilitator of secondary copy operations between client computing devices **102** and secondary storage devices **108**, but does not actually write the data to secondary storage device **108**. For instance, storage manager **140** (or media agent **144**) may instruct a client computing device **102** and secondary storage device **108** to communicate with one another directly. In such a case, client computing device **102** transmits data directly or via one or more intermediary components to secondary storage device **108** according to the received instructions, and vice versa. Media agent **144** may still receive, process, and/or maintain metadata related to the secondary copy operations, i.e., may continue to build and maintain index **153**. In these embodiments, payload data can flow through media agent **144** for the purposes of populating index **153**, but not for writing to secondary storage device **108**. Media agent **144** and/or other components such as storage manager **140** may in some cases incorporate additional functionality, such as data classification, content indexing, deduplication, encryption, compression, and the like. Further details regarding these and other functions are described below.

Distributed, Scalable Architecture

As described, certain functions of system **100** can be distributed amongst various physical and/or logical components. For instance, one or more of storage manager **140**, data agents **142**, and media agents **144** may operate on computing devices that are physically separate from one another. This architecture can provide a number of benefits. For instance, hardware and software design choices for each distributed component can be targeted to suit its particular function. The secondary computing devices **106** on which media agents **144** operate can be tailored for interaction with associated secondary storage devices **108** and provide fast index cache operation, among other specific tasks. Similarly, client computing device(s) **102** can be selected to effectively service applications **110** in order to efficiently produce and store primary data **112**.

Moreover, in some cases, one or more of the individual components of information management system **100** can be distributed to multiple separate computing devices. As one example, for large file systems where the amount of data stored in management database **146** is relatively large, database **146** may be migrated to or may otherwise reside on a specialized database server (e.g., an SQL server) separate from a server that implements the other functions of storage manager **140**. This distributed configuration can provide added protection because database **146** can be protected with standard database utilities (e.g., SQL log shipping or database replication) independent from other functions of storage manager **140**. Database **146** can be efficiently replicated to a remote site for use in the event of a disaster or other data loss at the primary site. Or database **146** can be replicated to another computing device within the same site, such as to a higher performance machine in the event that a storage manager host computing device can no longer service the needs of a growing system **100**.

The distributed architecture also provides scalability and efficient component utilization. FIG. **1D** shows an embodiment of information management system **100** including a plurality of client computing devices **102** and associated data agents **142** as well as a plurality of secondary storage computing devices **106** and associated media agents **144**. Additional components can be added or subtracted based on the evolving needs of system **100**. For instance, depending on where bottlenecks are identified, administrators can add additional client computing devices **102**, secondary storage computing devices **106**, and/or secondary storage devices **108**. Moreover, where multiple fungible components are available, load balancing can be implemented to dynamically address identified bottlenecks. As an example, storage manager **140** may dynamically select which media agents **144** and/or secondary storage devices **108** to use for storage operations based on a processing load analysis of media agents **144** and/or secondary storage devices **108**, respectively.

Where system **100** includes multiple media agents **144** (see, e.g., FIG. **1D**), a first media agent **144** may provide failover functionality for a second failed media agent **144**. In addition, media agents **144** can be dynamically selected to provide load balancing. Each client computing device **102** can communicate with, among other components, any of the media agents **144**, e.g., as directed by storage manager **140**. And each media agent **144** may communicate with, among other components, any of secondary storage devices **108**, e.g., as directed by storage manager **140**. Thus, operations can be routed to secondary storage devices **108** in a dynamic and highly flexible manner, to provide load balancing, failover, etc. Further examples of scalable systems capable of dynamic storage operations, load balancing, and failover are provided in U.S. Pat. No. 7,246,207.

While distributing functionality amongst multiple computing devices can have certain advantages, in other contexts it can be beneficial to consolidate functionality on the same computing device. In alternative configurations, certain components may reside and execute on the same computing device. As such, in other embodiments, one or more of the components shown in FIG. **1C** may be implemented on the same computing device. In one configuration, a storage manager **140**, one or more data agents **142**, and/or one or more media agents **144** are all implemented on the same computing device. In other embodiments, one or more data agents **142** and one or more media agents **144** are implemented on the same computing device, while storage manager **140** is implemented on a separate computing device, etc. without limitation.

Exemplary Types of Information Management Operations, Including Storage Operations

In order to protect and leverage stored data, system **100** can be configured to perform a variety of information management operations, which may also be referred to in some cases as storage management operations or storage operations. These operations can generally include (i) data movement operations, (ii) processing and data manipulation operations, and (iii) analysis, reporting, and management operations.

Data Movement Operations, Including Secondary Copy Operations

Data movement operations are generally storage operations that involve the copying or migration of data between different locations in system **100**. For example, data movement operations can include operations in which stored data is copied, migrated, or otherwise transferred from one or more first storage devices to one or more second storage devices, such as from primary storage device(s) **104** to secondary storage device(s) **108**, from secondary storage device(s) **108** to different secondary storage device(s) **108**, from secondary storage devices **108** to primary storage devices **104**, or from primary storage device(s) **104** to different primary storage device(s) **104**, or in some cases within the same primary storage device **104** such as within a storage array.

Data movement operations can include by way of example, backup operations, archive operations, information lifecycle management operations such as hierarchical storage management operations, replication operations (e.g., continuous data replication), snapshot operations, deduplication or single-instancing operations, auxiliary copy operations, disaster-recovery copy operations, and the like. As will be discussed, some of these operations do not necessarily create distinct copies. Nonetheless, some or all of these operations are generally referred to as "secondary copy operations" for simplicity, because they involve secondary copies. Data movement also comprises restoring secondary copies.

Backup Operations

A backup operation creates a copy of a version of primary data **112** at a particular point in time (e.g., one or more files or other data units). Each subsequent backup copy **116** (which is a form of secondary copy **116**) may be maintained independently of the first. A backup generally involves maintaining a version of the copied primary data **112** as well as backup copies **116**. Further, a backup copy in some embodiments is generally stored in a form that is different from the native format, e.g., a backup format. This contrasts to the version in primary data **112** which may instead be stored in a format native to the source application(s) **110**. In various cases, backup copies can be stored in a format in which the data is compressed, encrypted, deduplicated, and/or otherwise modified from the original native application format. For example, a backup copy may be stored in a compressed backup format that facilitates efficient long-term storage. Backup copies **116** can have relatively long retention periods as compared to primary data **112**, which is generally highly changeable. Backup copies **116** may be stored on media with slower retrieval times than primary storage device **104**. Some backup copies may have shorter retention periods than some other types of secondary copies **116**, such as archive copies (described below). Backups may be stored at an offsite location.

Backup operations can include full backups, differential backups, incremental backups, "synthetic full" backups, and/or creating a "reference copy." A full backup (or "standard full backup") in some embodiments is generally a complete image of the data to be protected. However, because full backup copies can consume a relatively large amount of storage, it can be useful to use a full backup copy as a baseline and only store changes relative to the full backup copy afterwards.

A differential backup operation (or cumulative incremental backup operation) tracks and stores changes that occurred since the last full backup. Differential backups can grow quickly in size, but can restore relatively efficiently because a restore can be completed in some cases using only the full backup copy and the latest differential copy.

An incremental backup operation generally tracks and stores changes since the most recent backup copy of any type, which can greatly reduce storage utilization. In some cases, however, restoring can be lengthy compared to full or differential backups because completing a restore operation may involve accessing a full backup in addition to multiple incremental backups.

Synthetic full backups generally consolidate data without directly backing up data from the client computing device. A synthetic full backup is created from the most recent full backup (i.e., standard or synthetic) and subsequent incremental and/or differential backups. The resulting synthetic full backup is identical to what would have been created had the last backup for the subclient been a standard full backup. Unlike standard full, incremental, and differential backups, however, a synthetic full backup does not actually transfer data from primary storage to the backup media, because it operates as a backup consolidator. A synthetic full backup extracts the index data of each participating subclient. Using this index data and the previously backed up user data images, it builds new full backup images (e.g., bitmaps), one for each subclient. The new backup images consolidate the index and user data stored in the related incremental, differential, and previous full backups into a synthetic backup file that fully represents the subclient (e.g., via pointers) but does not comprise all its constituent data.

Any of the above types of backup operations can be at the volume level, file level, or block level. Volume level backup operations generally involve copying of a data volume (e.g., a logical disk or partition) as a whole. In a file-level backup, information management system **100** generally tracks changes to individual files and includes copies of files in the backup copy. For block-level backups, files are broken into constituent blocks, and changes are tracked at the block level. Upon restore, system **100** reassembles the blocks into files in a transparent fashion. Far less data may actually be transferred and copied to secondary storage devices **108** during a file-level copy than a volume-level copy. Likewise, a block-level copy may transfer less data than a file-level copy, resulting in faster execution. However, restoring a relatively higher-granularity copy can result in longer restore times. For instance, when restoring a block-level copy, the process of locating and retrieving constituent blocks can sometimes take longer than restoring file-level backups.

A reference copy may comprise copy(ies) of selected objects from backed up data, typically to help organize data by keeping contextual information from multiple sources together, and/or help retain specific data for a longer period of time, such as for legal hold needs. A reference copy generally maintains data integrity, and when the data is restored, it may be viewed in the same format as the source data. In some embodiments, a reference copy is based on a specialized client, individual subclient and associated information management policies (e.g., storage policy, retention policy, etc.) that are administered within system **100**.

Archive Operations

Because backup operations generally involve maintaining a version of the copied primary data **112** and also maintaining backup copies in secondary storage device(s) **108**, they can consume significant storage capacity. To reduce storage consumption, an archive operation according to certain embodiments creates an archive copy **116** by both copying and removing source data. Or, seen another way, archive operations can involve moving some or all of the source data to the archive destination. Thus, data satisfying criteria for removal (e.g., data of a threshold age or size) may be removed from source storage. The source data may be primary data **112** or a secondary copy **116**, depending on the situation. As with backup copies, archive copies can be stored in a format in which the data is compressed, encrypted, deduplicated, and/or otherwise modified from the format of the original application or source copy. In addition, archive copies may be retained for relatively long periods of time (e.g., years) and, in some cases are never deleted. In certain embodiments, archive copies may be made and kept for extended periods in order to meet compliance regulations.

Archiving can also serve the purpose of freeing up space in primary storage device(s) **104** and easing the demand on computational resources on client computing device **102**. Similarly, when a secondary copy **116** is archived, the archive copy can therefore serve the purpose of freeing up space in the source secondary storage device(s) **108**. Examples of data archiving operations are provided in U.S. Pat. No. 7,107,298.

Snapshot Operations

Snapshot operations can provide a relatively lightweight, efficient mechanism for protecting data. From an end-user viewpoint, a snapshot may be thought of as an "instant" image of primary data **112** at a given point in time and may include state and/or status information relative to an application **110** that creates/manages primary data **112**. In one embodiment, a snapshot may generally capture the directory structure of an object in primary data **112** such as a file or volume or other data set at a particular moment in time and may also preserve file attributes and contents. A snapshot in some cases is created relatively quickly, e.g., substantially instantly, using a minimum amount of file space, but may still function as a conventional file system backup.

A "hardware snapshot" (or "hardware-based snapshot") operation occurs where a target storage device (e.g., a primary storage device **104** or a secondary storage device **108**) performs the snapshot operation in a self-contained fashion, substantially independently, using hardware, firmware and/or software operating on the storage device itself. For instance, the storage device may perform snapshot operations generally without intervention or oversight from any of the other components of the system **100**, e.g., a storage array may generate an "array-created" hardware snapshot and may also manage its storage, integrity, versioning, etc. In this manner, hardware snapshots can off-load other components of system **100** from snapshot processing. An array may receive a request from another component to take a snapshot and then proceed to execute the "hardware snapshot" operations autonomously, preferably reporting success to the requesting component.

A "software snapshot" (or "software-based snapshot") operation, on the other hand, occurs where a component in system **100** (e.g., client computing device **102**, etc.) implements a software layer that manages the snapshot operation via interaction with the target storage device. For instance, the component executing the snapshot management software layer may derive a set of pointers and/or data that represents the snapshot. The snapshot management software layer may then transmit the same to the target storage device, along with appropriate instructions for writing the snapshot. One example of a software snapshot product is Microsoft Volume Snapshot Service (VSS), which is part of the Microsoft Windows operating system.

Some types of snapshots do not actually create another physical copy of all the data as it existed at the particular point in time, but may simply create pointers that map files and directories to specific memory locations (e.g., to specific disk blocks) where the data resides as it existed at the particular point in time. For example, a snapshot copy may include a set of pointers derived from the file system or from an application. In some other cases, the snapshot may be created at the block-level, such that creation of the snapshot occurs without awareness of the file system. Each pointer points to a respective stored data block, so that collectively, the set of pointers reflect the storage location and state of the data object (e.g., file(s) or volume(s) or data set(s)) at the point in time when the snapshot copy was created.

An initial snapshot may use only a small amount of disk space needed to record a mapping or other data structure representing or otherwise tracking the blocks that correspond to the current state of the file system. Additional disk space is usually required only when files and directories change later on. Furthermore, when files change, typically only the pointers which map to blocks are copied, not the blocks themselves. For example, for "copy-on-write" snapshots, when a block changes in primary storage, the block is copied to secondary storage or cached in primary storage before the block is overwritten in primary storage, and the pointer to that block is changed to reflect the new location of that block. The snapshot mapping of file system data may also be updated to reflect the changed block(s) at that particular point in time. In some other cases, a snapshot includes a full physical copy of all or substantially all of the data represented by the snapshot. Further examples of snapshot operations are provided in U.S. Pat. No. 7,529,782. A snapshot copy in many cases can be made quickly and without significantly impacting primary computing resources because large amounts of data need not be copied or moved. In some embodiments, a snapshot may exist as a virtual file system, parallel to the actual file system. Users in some cases gain read-only access to the record of files and directories of the snapshot. By electing to restore primary data **112** from a snapshot taken at a given point in time, users may also return the current file system to the state of the file system that existed when the snapshot was taken.

Replication Operations

Replication is another type of secondary copy operation. Some types of secondary copies **116** periodically capture images of primary data **112** at particular points in time (e.g., backups, archives, and snapshots). However, it can also be useful for recovery purposes to protect primary data **112** in a more continuous fashion, by replicating primary data **112** substantially as changes occur. In some cases, a replication copy can be a mirror copy, for instance, where changes made to primary data **112** are mirrored or substantially immediately copied to another location (e.g., to secondary storage device(s) **108**). By copying each write operation to the replication copy, two storage systems are kept synchronized or substantially synchronized so that they are virtually identical at approximately the same time. Where entire disk volumes are mirrored, however, mirroring can require significant amount of storage space and utilizes a large amount of processing resources.

According to some embodiments, secondary copy operations are performed on replicated data that represents a recoverable state, or "known good state" of a particular application running on the source system. For instance, in certain embodiments, known good replication copies may be viewed as copies of primary data **112**. This feature allows the system to directly access, copy, restore, back up, or otherwise manipulate the replication copies as if they were the "live" primary data **112**. This can reduce access time, storage utilization, and impact on source applications **110**, among other benefits. Based on known good state information, system **100** can replicate sections of application data that represent a recoverable state rather than rote copying of blocks of data. Examples of replication operations (e.g., continuous data replication) are provided in U.S. Pat. No. 7,617,262.

Deduplication/Single-Instancing Operations

Deduplication or single-instance storage is useful to reduce the amount of non-primary data. For instance, some or all of the above-described secondary copy operations can involve deduplication in some fashion. New data is read, broken down into data portions of a selected granularity (e.g., sub-file level blocks, files, etc.), compared with corresponding portions that are already in secondary storage, and only new/changed portions are stored. Portions that already exist are represented as pointers to the already-stored data. Thus, a deduplicated secondary copy **116** may comprise actual data portions copied from primary data **112** and may further comprise pointers to already-stored data, which is generally more storage-efficient than a full copy.

In order to streamline the comparison process, system **100** may calculate and/or store signatures (e.g., hashes or cryptographically unique IDs) corresponding to the individual source data portions and compare the signatures to already-stored data signatures, instead of comparing entire data portions. In some cases, only a single instance of each data portion is stored, and deduplication operations may therefore be referred to interchangeably as "single-instancing" operations. Depending on the implementation, however, deduplication operations can store more than one instance of certain data portions, yet still significantly reduce stored-data redundancy. Depending on the embodiment, deduplication portions such as data blocks can be of fixed or variable length. Using variable length blocks can enhance deduplication by responding to changes in the data stream, but can involve more complex processing. In some cases, system **100** utilizes a technique for dynamically aligning deduplication blocks based on changing content in the data stream, as described in U.S. Pat. No. 8,364,652.

System **100** can deduplicate in a variety of manners at a variety of locations. For instance, in some embodiments, system **100** implements "target-side" deduplication by deduplicating data at the media agent **144** after being received from data agent **142**. In some such cases, media agents **144** are generally configured to manage the deduplication process. For instance, one or more of the media agents **144** maintain a corresponding deduplication database that stores deduplication information (e.g., data block signatures). Examples of such a configuration are provided in U.S. Pat. No. 9,020,900. Instead of or in combination with "target-side" deduplication, "source-side" (or "client-side") deduplication can also be performed, e.g., to reduce the amount of data to be transmitted by data agent **142** to media agent **144**. Storage manager **140** may communicate with other components within system **100** via network protocols and cloud service provider APIs to facilitate cloud-based deduplication/single instancing, as exemplified in U.S. Pat. No. 8,954,446. Some other deduplication/single instancing techniques are described in U.S. Pat. Pub. No. 2006/0224846 and in U.S. Pat. No. 9,098,495.

Information Lifecycle Management and Hierarchical Storage Management

In some embodiments, files and other data over their lifetime move from more expensive quick-access storage to less expensive slower-access storage. Operations associated with moving data through various tiers of storage are sometimes referred to as information lifecycle management (ILM) operations.

One type of ILM operation is a hierarchical storage management (HSM) operation, which generally automatically moves data between classes of storage devices, such as from high-cost to low-cost storage devices. For instance, an HSM operation may involve movement of data from primary storage devices **104** to secondary storage devices **108**, or between tiers of secondary storage devices **108**. With each tier, the storage devices may be progressively cheaper, have relatively slower access/restore times, etc. For example, movement of data between tiers may occur as data becomes less important over time. In some embodiments, an HSM operation is similar to archiving in that creating an HSM copy may (though not always) involve deleting some of the source data, e.g., according to one or more criteria related to the source data. For example, an HSM copy may include primary data **112** or a secondary copy **116** that exceeds a given size threshold or a given age threshold. Often, and unlike some types of archive copies, HSM data that is removed or aged from the source is replaced by a logical reference pointer or stub. The reference pointer or stub can be stored in the primary storage device **104** or other source storage device, such as a secondary storage device **108** to replace the deleted source data and to point to or otherwise indicate the new location in (another) secondary storage device **108**.

For example, files are generally moved between higher and lower cost storage depending on how often the files are accessed. When a user requests access to HSM data that has been removed or migrated, system **100** uses the stub to locate the data and can make recovery of the data appear transparent, even though the HSM data may be stored at a location different from other source data. In this manner, the data appears to the user (e.g., in file system browsing windows and the like) as if it still resides in the source location (e.g., in a primary storage device **104**). The stub may include metadata associated with the corresponding data, so that a file system and/or application can provide some information about the data object and/or a limited-functionality version (e.g., a preview) of the data object.

An HSM copy may be stored in a format other than the native application format (e.g., compressed, encrypted, deduplicated, and/or otherwise modified). In some cases, copies which involve the removal of data from source storage and the maintenance of stub or other logical reference information on source storage may be referred to generally as "online archive copies." On the other hand, copies which involve the removal of data from source storage without the maintenance of stub or other logical reference information on source storage may be referred to as "off-line archive copies." Examples of HSM and ILM techniques are provided in U.S. Pat. No. 7,343,453.

Auxiliary Copy Operations

An auxiliary copy is generally a copy of an existing secondary copy **116**. For instance, an initial secondary copy **116** may be derived from primary data **112** or from data residing in secondary storage subsystem **118**, whereas an auxiliary copy is generated from the initial secondary copy **116**. Auxiliary copies provide additional standby copies of data and may reside on different secondary storage devices **108** than the initial secondary copies **116**. Thus, auxiliary copies can be used for recovery purposes if initial secondary copies **116** become unavailable. Exemplary auxiliary copy techniques are described in further detail in U.S. Pat. No. 8,230,195.

Disaster-Recovery Copy Operations

System **100** may also make and retain disaster recovery copies, often as secondary, high-availability disk copies. System **100** may create secondary copies and store them at disaster recovery locations using auxiliary copy or replication operations, such as continuous data replication technologies. Depending on the particular data protection goals, disaster recovery locations can be remote from the client computing devices **102** and primary storage devices **104**, remote from some or all of the secondary storage devices **108**, or both.

Data Manipulation, Including Encryption and Compression

Data manipulation and processing may include encryption and compression as well as integrity marking and checking, formatting for transmission, formatting for storage, etc. Data may be manipulated "client-side" by data agent **142** as well as "target-side" by media agent **144** in the course of creating secondary copy **116**, or conversely in the course of restoring data from secondary to primary.

Encryption Operations

System **100** in some cases is configured to process data (e.g., files or other data objects, primary data **112**, secondary copies **116**, etc.), according to an appropriate encryption algorithm (e.g., Blowfish, Advanced Encryption Standard (AES), Triple Data Encryption Standard (3-DES), etc.) to limit access and provide data security. System **100** in some cases encrypts the data at the client level, such that client computing devices **102** (e.g., data agents **142**) encrypt the data prior to transferring it to other components, e.g., before sending the data to media agents **144** during a secondary copy operation. In such cases, client computing device **102** may maintain or have access to an encryption key or passphrase for decrypting the data upon restore. Encryption can also occur when media agent **144** creates auxiliary copies or archive copies. Encryption may be applied in creating a secondary copy **116** of a previously unencrypted secondary copy **116**, without limitation. In further embodiments, secondary storage devices **108** can implement built-in, high performance hardware-based encryption.

Compression Operations

Similar to encryption, system **100** may also or alternatively compress data in the course of generating a secondary copy **116**. Compression encodes information such that fewer bits are needed to represent the information as compared to the original representation. Compression techniques are well known in the art. Compression operations may apply one or more data compression algorithms. Compression may be applied in creating a secondary copy **116** of a previously uncompressed secondary copy, e.g., when making archive copies or disaster recovery copies. The use of compression may result in metadata that specifies the nature of the compression, so that data may be uncompressed on restore if appropriate.

Data Analysis, Reporting, and Management Operations

Data analysis, reporting, and management operations can differ from data movement operations in that they do not necessarily involve copying, migration or other transfer of data between different locations in the system. For instance, data analysis operations may involve processing (e.g., offline processing) or modification of already stored primary data **112** and/or secondary copies **116**. However, in some embodiments data analysis operations are performed in conjunction with data movement operations. Some data analysis operations include content indexing operations and classification operations which can be useful in leveraging data under management to enhance search and other features.

Classification Operations/Content Indexing

In some embodiments, information management system **100** analyzes and indexes characteristics, content, and metadata associated with primary data **112** ("online content indexing") and/or secondary copies **116** ("off-line content indexing"). Content indexing can identify files or other data objects based on content (e.g., user-defined keywords or phrases, other keywords/phrases that are not defined by a user, etc.), and/or metadata (e.g., email metadata such as "to," "from," "cc," "bcc," attachment name, received time, etc.). Content indexes may be searched, and search results may be restored.

System **100** generally organizes and catalogues the results into a content index, which may be stored within media agent database **152**, for example. The content index can also include the storage locations of or pointer references to indexed data in primary data **112** and/or secondary copies **116**. Results may also be stored elsewhere in system **100** (e.g., in primary storage device **104** or in secondary storage device **108**). Such content index data provides storage manager **140** or other components with an efficient mechanism for locating primary data **112** and/or secondary copies **116** of data objects that match particular criteria, thus greatly increasing the search speed capability of system **100**. For instance, search criteria can be specified by a user through user interface **158** of storage manager **140**. Moreover, when system **100** analyzes data and/or metadata in secondary copies **116** to create an "off-line content index," this operation has no significant impact on the performance of client computing devices **102** and thus does not take a toll on the production environment. Examples of content indexing techniques are provided in U.S. Pat. No. 8,170,995.

One or more components, such as a content index engine, can be configured to scan data and/or associated metadata for classification purposes to populate a database (or other data structure) of information, which can be referred to as a "data classification database" or a "metabase." Depending on the embodiment, the data classification database(s) can be organized in a variety of different ways, including centralization, logical sub-divisions, and/or physical sub-divisions. For instance, one or more data classification databases may be associated with different subsystems or tiers within system **100**. As an example, there may be a first metabase associated with primary storage subsystem **117** and a second metabase associated with secondary storage subsystem **118**. In other cases, metabase(s) may be associated with individual components, e.g., client computing devices **102** and/or media agents **144**. In some embodiments, a data classification database may reside as one or more data structures within management database **146**, may be otherwise associated with storage manager **140**, and/or may reside as a separate component. In some cases, metabase(s) may be included in separate database(s) and/or on separate storage device(s) from primary data **112** and/or secondary copies **116**, such that operations related to the metabase(s) do not significantly impact performance on other components of system **100**. In other cases, metabase(s) may be stored along with primary data **112** and/or secondary copies **116**. Files or other data objects can be associated with identifiers (e.g., tag entries, etc.) to facilitate searches of stored data objects. Among a number of other benefits, the metabase can also allow efficient, automatic identification of files or other data objects to associate with secondary copy or other information management operations. For instance, a metabase can dramatically improve the speed with which system **100** can search through and identify data as compared to other approaches that involve scanning an entire file system. Examples of metabases and data classification operations are provided in U.S. Pat. Nos. 7,734,669 and 7,747,579.

Management and Reporting Operations

Certain embodiments leverage the integrated ubiquitous nature of system **100** to provide useful system-wide management and reporting. Operations management can generally include monitoring and managing the health and performance of system **100** by, without limitation, performing error tracking, generating granular storage/performance metrics (e.g., job success/failure information, deduplication efficiency, etc.), generating storage modeling and costing information, and the like. As an example, storage manager **140** or another component in system **100** may analyze traffic patterns and suggest and/or automatically route data to minimize congestion. In some embodiments, the system can generate predictions relating to storage operations or storage operation information. Such predictions, which may be based on a trending analysis, may predict various network operations or resource usage, such as network traffic levels, storage media use, use of bandwidth of communication links, use of media agent components, etc. Further examples of traffic analysis, trend analysis, prediction generation, and the like are described in U.S. Pat. No. 7,343,453.

In some configurations having a hierarchy of storage operation cells, a master storage manager **140** may track the status of subordinate cells, such as the status of jobs, system components, system resources, and other items, by communicating with storage managers **140** (or other components) in the respective storage operation cells. Moreover, the master storage manager **140** may also track status by receiving periodic status updates from the storage managers **140** (or other components) in the respective cells regarding jobs, system components, system resources, and other items. In some embodiments, a master storage manager **140** may store status information and other information regarding its associated storage operation cells and other system information in its management database **146** and/or index **150** (or in another location). The master storage manager **140** or other component may also determine whether certain storage-related or other criteria are satisfied, and may perform an action or trigger event (e.g., data migration) in response to the criteria being satisfied, such as where a storage threshold is met for a particular volume, or where inadequate protection exists for certain data. For instance, data from one or more storage operation cells is used to dynamically and automatically mitigate recognized risks, and/or to advise users of risks or suggest actions to mitigate these risks. For example, an information management policy may specify certain requirements (e.g., that a storage device should maintain a certain amount of free space, that secondary copies should occur at a particular interval, that data should be aged and migrated to other storage after a particular period, that data on a secondary volume should always have a certain level of availability and be restorable within a given time period, that data on a secondary volume may be mirrored or otherwise migrated to a specified number of other volumes, etc.). If a risk condition or other criterion is triggered, the system may notify the user of these conditions and may suggest (or automatically implement) a mitigation action to address the risk. For example, the system may indicate that data from a primary copy **112** should be migrated to a secondary storage device **108** to free up space on primary storage device **104**. Examples of the use of risk factors and other triggering criteria are described in U.S. Pat. No. 7,343,453.

In some embodiments, system **100** may also determine whether a metric or other indication satisfies particular storage criteria sufficient to perform an action. For example, a storage policy or other definition might indicate that a storage manager **140** should initiate a particular action if a storage metric or other indication drops below or otherwise fails to satisfy specified criteria such as a threshold of data protection. In some embodiments, risk factors may be quantified into certain measurable service or risk levels. For example, certain applications and associated data may be considered to be more important relative to other data and services. Financial compliance data, for example, may be of greater importance than marketing materials, etc. Network administrators may assign priority values or "weights" to certain data and/or applications corresponding to the relative importance. The level of compliance of secondary copy operations specified for these applications may also be assigned a certain value. Thus, the health, impact, and overall importance of a service may be determined, such as by measuring the compliance value and calculating the product of the priority value and the compliance value to determine the "service level" and comparing it to certain operational thresholds to determine whether it is acceptable. Further examples of the service level determination are provided in U.S. Pat. No. 7,343,453.

System **100** may additionally calculate data costing and data availability associated with information management operation cells. For instance, data received from a cell may be used in conjunction with hardware-related information and other information about system elements to determine the cost of storage and/or the availability of particular data. Exemplary information generated could include how fast a particular department is using up available storage space, how long data would take to recover over a particular pathway from a particular secondary storage device, costs over time, etc. Moreover, in some embodiments, such information may be used to determine or predict the overall cost associated with the storage of certain information. The cost associated with hosting a certain application may be based, at least in part, on the type of media on which the data resides, for example. Storage devices may be assigned to a particular cost categories, for example. Further examples of costing techniques are described in U.S. Pat. No. 7,343,453.

Any of the above types of information (e.g., information related to trending, predictions, job, cell or component status, risk, service level, costing, etc.) can generally be provided to users via user interface **158** in a single integrated view or console (not shown). Report types may include: scheduling, event management, media management and data aging. Available reports may also include backup history, data aging history, auxiliary copy history, job history, library and drive, media in library, restore history, and storage policy, etc., without limitation. Such reports may be specified and created at a certain point in time as a system analysis, forecasting, or provisioning tool. Integrated reports may also be generated that illustrate storage and performance metrics, risks and storage costing information. Moreover, users may create their own reports based on specific needs. User interface **158** can include an option to graphically depict the various components in the system using appropriate icons. As one example, user interface **158** may provide a graphical depiction of primary storage devices **104**, secondary storage devices **108**, data agents **142** and/or media agents **144**, and their relationship to one another in system **100**.

In general, the operations management functionality of system **100** can facilitate planning and decision-making. For example, in some embodiments, a user may view the status of some or all jobs as well as the status of each component of information management system **100**. Users may then plan and make decisions based on this data. For instance, a user may view high-level information regarding secondary copy operations for system **100**, such as job status, component status, resource status (e.g., communication pathways, etc.), and other information. The user may also drill down or use other means to obtain more detailed information regarding a particular component, job, or the like. Further examples are provided in U.S. Pat. No. 7,343,453.

System **100** can also be configured to perform system-wide e-discovery operations in some embodiments. In general, e-discovery operations provide a unified collection and search capability for data in the system, such as data stored in secondary storage devices **108** (e.g., backups, archives, or other secondary copies **116**). For example, system **100** may construct and maintain a virtual repository for data stored in system **100** that is integrated across source applications **110**, different storage device types, etc. According to some embodiments, e-discovery utilizes other techniques described herein, such as data classification and/or content indexing.

Information Management Policies

An information management policy **148** can include a data structure or other information source that specifies a set of parameters (e.g., criteria and rules) associated with secondary copy and/or other information management operations.

One type of information management policy **148** is a "storage policy." According to certain embodiments, a storage policy generally comprises a data structure or other information source that defines (or includes information sufficient to determine) a set of preferences or other criteria for performing information management operations. Storage policies can include one or more of the following: (1) what data will be associated with the storage policy, e.g., subclient; (2) a destination to which the data will be stored; (3) datapath information specifying how the data will be communicated to the destination; (4) the type of secondary copy operation to be performed; and (5) retention information specifying how long the data will be retained at the destination (see, e.g., FIG. 1E). Data associated with a storage policy can be logically organized into subclients, which may represent primary data **112** and/or secondary copies **116**. A subclient may represent static or dynamic associations of portions of a data volume. Subclients may represent mutually exclusive portions. Thus, in certain embodiments, a portion of data may be given a label and the association is stored as a static entity in an index, database or other storage location. Subclients may also be used as an effective administrative scheme of organizing data according to data type, department within the enterprise, storage preferences, or the like. Depending on the configuration, subclients can correspond to files, folders, virtual machines, databases, etc. In one exemplary scenario, an administrator may find it preferable to separate e-mail data from financial data using two different subclients.

A storage policy can define where data is stored by specifying a target or destination storage device (or group of storage devices). For instance, where the secondary storage device **108** includes a group of disk libraries, the storage policy may specify a particular disk library for storing the subclients associated with the policy. As another example, where the secondary storage devices **108** include one or more tape libraries, the storage policy may specify a particular tape library for storing the subclients associated with the storage policy, and may also specify a drive pool and a tape pool defining a group of tape drives and a group of tapes, respectively, for use in storing the subclient data. While information in the storage policy can be statically assigned in some cases, some or all of the information in the storage policy can also be dynamically determined based on criteria set forth in the storage policy. For instance, based on such criteria, a particular destination storage device(s) or other parameter of the storage policy may be determined based on characteristics associated with the data involved in a particular secondary copy operation, device availability (e.g., availability of a secondary storage device **108** or a media agent **144**), network status and conditions (e.g., identified bottlenecks), user credentials, and the like.

Datapath information can also be included in the storage policy. For instance, the storage policy may specify network pathways and components to utilize when moving the data to the destination storage device(s). In some embodiments, the storage policy specifies one or more media agents **144** for conveying data associated with the storage policy between the source and destination. A storage policy can also specify the type(s) of associated operations, such as backup, archive, snapshot, auxiliary copy, or the like. Furthermore, retention parameters can specify how long the resulting secondary copies **116** will be kept (e.g., a number of days, months, years, etc.), perhaps depending on organizational needs and/or compliance criteria.

When adding a new client computing device **102**, administrators can manually configure information management policies **148** and/or other settings, e.g., via user interface **158**. However, this can be an involved process resulting in delays, and it may be desirable to begin data protection operations quickly, without awaiting human intervention. Thus, in some embodiments, system **100** automatically applies a default configuration to client computing device **102**. As one example, when one or more data agent(s) **142** are installed on a client computing device **102**, the installation script may register the client computing device **102** with storage manager **140**, which in turn applies the default configuration to the new client computing device **102**. In this manner, data protection operations can begin substantially immediately. The default configuration can include a default storage policy, for example, and can specify any appropriate information sufficient to begin data protection operations. This can include a type of data protection operation, scheduling information, a target secondary storage device **108**, data path information (e.g., a particular media agent **144**), and the like.

Another type of information management policy **148** is a "scheduling policy," which specifies when and how often to perform operations. Scheduling parameters may specify with what frequency (e.g., hourly, weekly, daily, event-based, etc.) or under what triggering conditions secondary copy or other information management operations are to take place. Scheduling policies in some cases are associated with particular components, such as a subclient, client computing device **102**, and the like.

Another type of information management policy **148** is an "audit policy" (or "security policy"), which comprises preferences, rules and/or criteria that protect sensitive data in system **100**. For example, an audit policy may define "sensitive objects" which are files or data objects that contain particular keywords (e.g., "confidential," or "privileged") and/or are associated with particular keywords (e.g., in metadata) or particular flags (e.g., in metadata identifying a document or email as personal, confidential, etc.). An audit policy may further specify rules for handling sensitive objects. As an example, an audit policy may require that a reviewer approve the transfer of any sensitive objects to a cloud storage site, and that if approval is denied for a particular sensitive object, the sensitive object should be transferred to a local primary storage device **104** instead. To facilitate this approval, the audit policy may further specify how a secondary storage computing device **106** or other system component should notify a reviewer that a sensitive object is slated for transfer.

Another type of information management policy **148** is a "provisioning policy," which can include preferences, priorities, rules, and/or criteria that specify how client computing devices **102** (or groups thereof) may utilize system resources, such as available storage on cloud storage and/or network bandwidth. A provisioning policy specifies, for example, data quotas for particular client computing devices **102** (e.g., a number of gigabytes that can be stored monthly, quarterly or annually). Storage manager **140** or other components may enforce the provisioning policy. For instance, media agents **144** may enforce the policy when transferring data to secondary storage devices **108**. If a client computing device **102** exceeds a quota, a budget for the client computing device **102** (or associated department) may be adjusted accordingly or an alert may trigger.

While the above types of information management policies **148** are described as separate policies, one or more of these can be generally combined into a single information management policy **148**. For instance, a storage policy may also include or otherwise be associated with one or more scheduling, audit, or provisioning policies or operational parameters thereof. Moreover, while storage policies are typically associated with moving and storing data, other policies may be associated with other types of information management operations. The following is a non-exhaustive list of items that information management policies **148** may specify:

- schedules or other timing information, e.g., specifying when and/or how often to perform information management operations;
- the type of secondary copy **116** and/or copy format (e.g., snapshot, backup, archive, HSM, etc.);
- a location or a class or quality of storage for storing secondary copies **116** (e.g., one or more particular secondary storage devices **108**);
- preferences regarding whether and how to encrypt, compress, deduplicate, or otherwise modify or transform secondary copies **116**;
- which system components and/or network pathways (e.g., preferred media agents **144**) should be used to perform secondary storage operations;
- resource allocation among different computing devices or other system components used in performing information management operations (e.g., bandwidth allocation, available storage capacity, etc.);
- whether and how to synchronize or otherwise distribute files or other data objects across multiple computing devices or hosted services; and
- retention information specifying the length of time primary data **112** and/or secondary copies **116** should be retained, e.g., in a particular class or tier of storage devices, or within the system **100**.

Information management policies **148** can additionally specify or depend on historical or current criteria that may be used to determine which rules to apply to a particular data object, system component, or information management operation, such as:

- frequency with which primary data **112** or a secondary copy **116** of a data object or metadata has been or is predicted to be used, accessed, or modified;
- time-related factors (e.g., aging information such as time since the creation or modification of a data object);
- deduplication information (e.g., hashes, data blocks, deduplication block size, deduplication efficiency or other metrics);
- an estimated or historic usage or cost associated with different components (e.g., with secondary storage devices **108**);
- the identity of users, applications **110**, client computing devices **102** and/or other computing devices that created, accessed, modified, or otherwise utilized primary data **112** or secondary copies **116**;
- a relative sensitivity (e.g., confidentiality, importance) of a data object, e.g., as determined by its content and/or metadata;
- the current or historical storage capacity of various storage devices;
- the current or historical network capacity of network pathways connecting various components within the storage operation cell;
- access control lists or other security information; and
- the content of a particular data object (e.g., its textual content) or of metadata associated with the data object.

Exemplary Storage Policy and Secondary Copy Operations

FIG. **1**E includes a data flow diagram depicting performance of secondary copy operations by an embodiment of information management system **100**, according to an exemplary storage policy **148**A. System **100** includes a storage manager **140**, a client computing device **102** having a file system data agent **142**A and an email data agent **142**B operating thereon, a primary storage device **104**, two media agents **144**A, **144**B, and two secondary storage devices **108**: a disk library **108**A and a tape library **108**B. As shown, primary storage device **104** includes primary data **112**A, which is associated with a logical grouping of data associated with a file system ("file system subclient"), and primary data **112**B, which is a logical grouping of data associated with email ("email subclient"). The techniques described with respect to FIG. **1**E can be utilized in conjunction with data that is otherwise organized as well.

As indicated by the dashed box, the second media agent **144**B and tape library **108**B are "off-site," and may be remotely located from the other components in system **100** (e.g., in a different city, office building, etc.). Indeed, "off-site" may refer to a magnetic tape located in remote storage, which must be manually retrieved and loaded into a tape drive to be read. In this manner, information stored on the tape library **108**B may provide protection in the event of a disaster or other failure at the main site(s) where data is stored.

The file system subclient **112**A in certain embodiments generally comprises information generated by the file system and/or operating system of client computing device **102**, and can include, for example, file system data (e.g., regular files, file tables, mount points, etc.), operating system data (e.g., registries, event logs, etc.), and the like. The e-mail subclient **112**B can include data generated by an e-mail application operating on client computing device **102**, e.g., mailbox information, folder information, emails, attachments, associated database information, and the like. As described above, the subclients can be logical containers, and the data included in the corresponding primary data **112**A and **112**B may or may not be stored contiguously.

The exemplary storage policy **148**A includes backup copy preferences or rule set **160**, disaster recovery copy preferences or rule set **162**, and compliance copy preferences or rule set **164**. Backup copy rule set **160** specifies that it is associated with file system subclient **166** and email subclient **168**. Each of subclients **166** and **168** are associated with the particular client computing device **102**. Backup copy rule set **160** further specifies that the backup operation will be written to disk library **108**A and designates a particular media agent **144**A to convey the data to disk library **108**A. Finally, backup copy rule set **160** specifies that backup copies created according to rule set **160** are scheduled to be generated hourly and are to be retained for 30 days. In some other embodiments, scheduling information is not included in storage policy **148**A and is instead specified by a separate scheduling policy.

Disaster recovery copy rule set **162** is associated with the same two subclients **166** and **168**. However, disaster recovery copy rule set **162** is associated with tape library **108**B, unlike backup copy rule set **160**. Moreover, disaster recovery copy rule set **162** specifies that a different media agent, namely **144**B, will convey data to tape library **108**B. Disaster recovery copies created according to rule set **162** will be retained for 60 days and will be generated daily. Disaster recovery copies generated according to disaster recovery copy rule set **162** can provide protection in the event of a disaster or other catastrophic data loss that would affect the backup copy **116**A maintained on disk library **108**A.

Compliance copy rule set **164** is only associated with the email subclient **168**, and not the file system subclient **166**. Compliance copies generated according to compliance copy rule set **164** will therefore not include primary data **112**A from the file system subclient **166**. For instance, the organization may be under an obligation to store and maintain copies of email data for a particular period of time (e.g., 10 years) to comply with state or federal regulations, while similar regulations do not apply to file system data. Compliance copy rule set **164** is associated with the same tape library **108**B and media agent **144**B as disaster recovery copy rule set **162**, although a different storage device or media agent could be used in other embodiments. Finally, compliance copy rule set **164** specifies that the copies it governs will be generated quarterly and retained for 10 years.

Secondary Copy Jobs

A logical grouping of secondary copy operations governed by a rule set and being initiated at a point in time may be referred to as a "secondary copy job" (and sometimes may be called a "backup job," even though it is not necessarily limited to creating only backup copies). Secondary copy jobs may be initiated on demand as well. Steps **1-9** below illustrate three secondary copy jobs based on storage policy **148**A.

Referring to FIG. **1**E, at step **1**, storage manager **140** initiates a backup job according to the backup copy rule set **160**, which logically comprises all the secondary copy operations necessary to effectuate rules **160** in storage policy **148**A every hour, including steps **1-4** occurring hourly. For instance, a scheduling service running on storage manager **140** accesses backup copy rule set **160** or a separate scheduling policy associated with client computing device **102** and initiates a backup job on an hourly basis. Thus, at the scheduled time, storage manager **140** sends instructions to client computing device **102** (i.e., to both data agent **142**A and data agent **142**B) to begin the backup job.

At step **2**, file system data agent **142**A and email data agent **142**B on client computing device **102** respond to instructions from storage manager **140** by accessing and processing the respective subclient primary data **112**A and **112**B involved in the backup copy operation, which can be found in primary storage device **104**. Because the secondary copy operation is a backup copy operation, the data agent(s) **142**A, **142**B may format the data into a backup format or otherwise process the data suitable for a backup copy.

At step **3**, client computing device **102** communicates the processed file system data (e.g., using file system data agent **142**A) and the processed email data (e.g., using email data agent **142**B) to the first media agent **144**A according to backup copy rule set **160**, as directed by storage manager **140**. Storage manager **140** may further keep a record in management database **146** of the association between media agent **144**A and one or more of: client computing device **102**, file system subclient **112**A, file system data agent **142**A, email subclient **112**B, email data agent **142**B, and/or backup copy **116**A.

The target media agent **144**A receives the data-agent-processed data from client computing device **102**, and at step **4** generates and conveys backup copy **116**A to disk library **108**A to be stored as backup copy **116**A, again at the direction of storage manager **140** and according to backup copy rule set **160**. Media agent **144**A can also update its index **153** to include data and/or metadata related to backup copy **116**A, such as information indicating where the backup copy **116**A resides on disk library **108**A, where the email copy resides, where the file system copy resides, data and metadata for cache retrieval, etc. Storage manager **140** may similarly update its index **150** to include information relating to the secondary copy operation, such as information relating to the type of operation, a physical location associated with one or more copies created by the operation, the time the operation was performed, status information relating to the operation, the components involved in the operation, and the like. In some cases, storage manager **140** may update its index **150** to include some or all of the information stored in index **153** of media agent **144**A. At this point, the backup job may be considered complete. After the 30-day retention period expires, storage manager **140** instructs media agent **144**A to delete backup copy **116**A from disk library **108**A and indexes **150** and/or **153** are updated accordingly.

At step **5**, storage manager **140** initiates another backup job for a disaster recovery copy according to the disaster recovery rule set **162**. Illustratively this includes steps **5-7** occurring daily for creating disaster recovery copy **116**B. Illustratively, and by way of illustrating the scalable aspects and off-loading principles embedded in system **100**, disaster recovery copy **116**B is based on backup copy **116**A and not on primary data **112**A and **112**B.

At step **6**, illustratively based on instructions received from storage manager **140** at step **5**, the specified media agent **1446** retrieves the most recent backup copy **116**A from disk library **108**A.

At step **7**, again at the direction of storage manager **140** and as specified in disaster recovery copy rule set **162**, media agent **144**B uses the retrieved data to create a disaster recovery copy **1166** and store it to tape library **1086**. In some cases, disaster recovery copy **116**B is a direct, mirror copy of backup copy **116**A, and remains in the backup format. In other embodiments, disaster recovery copy **116**B may be further compressed or encrypted, or may be generated in some other manner, such as by using primary data **112**A and **1126** from primary storage device **104** as sources. The disaster recovery copy operation is initiated once a day and disaster recovery copies **1166** are deleted after 60 days; indexes **153** and/or **150** are updated accordingly when/after each information management operation is executed and/or completed. The present backup job may be considered completed.

At step **8**, storage manager **140** initiates another backup job according to compliance rule set **164**, which performs steps **8-9** quarterly to create compliance copy **116**C. For instance, storage manager **140** instructs media agent **144**B to create compliance copy **116**C on tape library **1086**, as specified in the compliance copy rule set **164**.

At step **9** in the example, compliance copy **116**C is generated using disaster recovery copy **1166** as the source. This is efficient, because disaster recovery copy resides on the same secondary storage device and thus no network resources are required to move the data. In other embodiments, compliance copy **116**C is instead generated using primary data **112**B corresponding to the email subclient or using backup copy **116**A from disk library **108**A as source data. As specified in the illustrated example, compliance copies **116**C are created quarterly, and are deleted after ten years, and indexes **153** and/or **150** are kept up-to-date accordingly.

Exemplary Applications of Storage Policies—Information Governance Policies and Classification

Again referring to FIG. **1**E, storage manager **140** may permit a user to specify aspects of storage policy **148**A. For example, the storage policy can be modified to include information governance policies to define how data should be managed in order to comply with a certain regulation or business objective. The various policies may be stored, for example, in management database **146**. An information governance policy may align with one or more compliance tasks that are imposed by regulations or business requirements. Examples of information governance policies might include a Sarbanes-Oxley policy, a HIPAA policy, an electronic discovery (e-discovery) policy, and so on.

Information governance policies allow administrators to obtain different perspectives on an organization's online and offline data, without the need for a dedicated data silo created solely for each different viewpoint. As described previously, the data storage systems herein build an index that reflects the contents of a distributed data set that spans numerous clients and storage devices, including both primary data and secondary copies, and online and offline copies. An organization may apply multiple information governance policies in a top-down manner over that unified data set and indexing schema in order to view and manipulate the data set through different lenses, each of which is adapted to a particular compliance or business goal. Thus, for example, by applying an e-discovery policy and a Sarbanes-Oxley policy, two different groups of users in an organization can conduct two very different analyses of the same underlying physical set of data/copies, which may be distributed throughout the information management system.

An information governance policy may comprise a classification policy, which defines a taxonomy of classification terms or tags relevant to a compliance task and/or business objective. A classification policy may also associate a defined tag with a classification rule. A classification rule defines a particular combination of criteria, such as users who have created, accessed or modified a document or data object; file or application types; content or metadata keywords; clients or storage locations; dates of data creation and/or access; review status or other status within a workflow (e.g., reviewed or un-reviewed); modification times or types of modifications; and/or any other data attributes in any combination, without limitation. A classification rule may also be defined using other classification tags in the taxonomy. The various criteria used to define a classification rule may be combined in any suitable fashion, for example, via Boolean operators, to define a complex classification rule. As an example, an e-discovery classification policy might define a classification tag "privileged" that is associated with documents or data objects that (1) were created or modified by legal department staff, or (2) were sent to or received from outside counsel via email, or (3) contain one of the following keywords: "privileged" or "attorney" or "counsel," or other like terms. Accordingly, all these documents or data objects will be classified as "privileged."

One specific type of classification tag, which may be added to an index at the time of indexing, is an "entity tag." An entity tag may be, for example, any content that matches a defined data mask format. Examples of entity tags might include, e.g., social security numbers (e.g., any numerical content matching the formatting mask XXX-XX-XXXX), credit card numbers (e.g., content having a 13-16 digit string of numbers), SKU numbers, product numbers, etc. A user may define a classification policy by indicating criteria, parameters or descriptors of the policy via a graphical user interface, such as a form or page with fields to be filled in, pull-down menus or entries allowing one or more of several options to be selected, buttons, sliders, hypertext links or other known user interface tools for receiving user input, etc. For example, a user may define certain entity tags, such as a particular product number or project ID. In some implementations, the classification policy can be implemented using cloud-based techniques. For example, the storage devices may be cloud storage devices, and the storage manager **140** may execute cloud service provider API over a network to classify data stored on cloud storage devices.

Restore Operations from Secondary Copies

While not shown in FIG. **1**E, at some later point in time, a restore operation can be initiated involving one or more of secondary copies **116**A, **116**B, and **116**C. A restore operation logically takes a selected secondary copy **116**, reverses the effects of the secondary copy operation that created it, and stores the restored data to primary storage where a client computing device **102** may properly access it as primary data. A media agent **144** and an appropriate data agent **142** (e.g., executing on the client computing device **102**) perform the tasks needed to complete a restore operation. For example, data that was encrypted, compressed, and/or deduplicated in the creation of secondary copy **116** will be correspondingly rehydrated (reversing deduplication), uncompressed, and unencrypted into a format appropriate to primary data. Metadata stored within or associated with the secondary copy **116** may be used during the restore operation. In general, restored data should be indistinguishable from other primary data **112**. Preferably, the restored data has fully regained the native format that may make it immediately usable by application **110**.

As one example, a user may manually initiate a restore of backup copy **116**A, e.g., by interacting with user interface **158** of storage manager **140** or with a web-based console with access to system **100**. Storage manager **140** may accesses data in its index **150** and/or management database **146** (and/or the respective storage policy **148**A) associated with the selected backup copy **116**A to identify the appropriate media agent **144**A and/or secondary storage device **108**A where the secondary copy resides. The user may be presented with a representation (e.g., stub, thumbnail, listing, etc.) and metadata about the selected secondary copy, in order to determine whether this is the appropriate copy to be restored, e.g., date that the original primary data was created. Storage manager **140** will then instruct media agent **144**A and an appropriate data agent **142** on the target client computing device **102** to restore secondary copy **116**A to primary storage device **104**. A media agent may be selected for use in the restore operation based on a load balancing algorithm, an availability based algorithm, or other criteria. The selected media agent, e.g., **144**A, retrieves secondary copy **116**A from disk library **108**A. For instance, media agent **144**A may access its index **153** to identify a location of backup copy **116**A on disk library **108**A, or may access location information residing on disk library **108**A itself.

In some cases, a backup copy **116**A that was recently created or accessed, may be cached to speed up the restore operation. In such a case, media agent **144**A accesses a cached version of backup copy **116**A residing in index **153**, without having to access disk library **108**A for some or all of the data. Once it has retrieved backup copy **116**A, the media agent **144**A communicates the data to the requesting client computing device **102**. Upon receipt, file system data agent **142**A and email data agent **142**B may unpack (e.g., restore from a backup format to the native application format) the data in backup copy **116**A and restore the unpackaged data to primary storage device **104**. In general, secondary copies **116** may be restored to the same volume or folder in primary storage device **104** from which the secondary copy was derived; to another storage location or client computing device **102**; to shared storage, etc. In some cases, the data may be restored so that it may be used by an application **110** of a different version/vintage from the application that created the original primary data **112**.

Exemplary Secondary Copy Formatting

The formatting and structure of secondary copies **116** can vary depending on the embodiment. In some cases, secondary copies **116** are formatted as a series of logical data units or "chunks" (e.g., 512 MB, 1 GB, 2 GB, 4 GB, or 8 GB chunks). This can facilitate efficient communication and writing to secondary storage devices **108**, e.g., according to resource availability. For example, a single secondary copy **116** may be written on a chunk-by-chunk basis to one or more secondary storage devices **108**. In some cases, users can select different chunk sizes, e.g., to improve throughput to tape storage devices. Generally, each chunk can include a header and a payload. The payload can include files (or other data units) or subsets thereof included in the chunk, whereas the chunk header generally includes metadata relating to the chunk, some or all of which may be derived from the payload. For example, during a secondary copy operation, media agent **144**, storage manager **140**, or other component may divide files into chunks and generate headers for each chunk by processing the files. Headers can include a variety of information such as file and/or volume identifier(s), offset(s), and/or other information associated with the payload data items, a chunk sequence number, etc. Importantly, in addition to being stored with secondary copy **116** on secondary storage device **108**, chunk headers can also be stored to index **153** of the associated media agent(s) **144** and/or to index **150** associated with storage manager **140**. This can be useful for providing faster processing of secondary copies **116** during browsing, restores, or other operations. In some cases, once a chunk is successfully transferred to a secondary storage device **108**, the secondary storage device **108** returns an indication of receipt, e.g., to media agent **144** and/or storage manager **140**, which may update their respective indexes **153**, **150** accordingly. During restore, chunks may be processed (e.g., by media agent **144**) according to the information in the chunk header to reassemble the files.

Data can also be communicated within system **100** in data channels that connect client computing devices **102** to secondary storage devices **108**. These data channels can be referred to as "data streams," and multiple data streams can be employed to parallelize an information management operation, improving data transfer rate, among other advantages. Example data formatting techniques including techniques involving data streaming, chunking, and the use of other data structures in creating secondary copies are described in U.S. Pat. Nos. 7,315,923, 8,156,086, and 8,578,120.

FIGS. **1**F and **1**G are diagrams of example data streams **170** and **171**, respectively, which may be employed for performing information management operations. Referring to FIG. **1**F, data agent **142** forms data stream **170** from source data associated with a client computing device **102** (e.g., primary data **112**). Data stream **170** is composed of multiple pairs of stream header **172** and stream data (or stream payload) **174**. Data streams **170** and **171** shown in the illustrated example are for a single-instanced storage operation, and a stream payload **174** therefore may include both single-instance (SI) data and/or non-SI data. A stream header **172** includes metadata about the stream payload **174**. This metadata may include, for example, a length of the stream payload **174**, an indication of whether the stream payload **174** is encrypted, an indication of whether the stream payload **174** is compressed, an archive file identifier (ID), an indication of whether the stream payload **174** is single instanceable, and an indication of whether the stream payload **174** is a start of a block of data.

Referring to FIG. **1**G, data stream **171** has the stream header **172** and stream payload **174** aligned into multiple data blocks. In this example, the data blocks are of size 64 KB. The first two stream header **172** and stream payload **174** pairs comprise a first data block of size 64 KB. The first stream header **172** indicates that the length of the succeeding stream payload **174** is 63 KB and that it is the start of a data block. The next stream header **172** indicates that the succeeding stream payload **174** has a length of 1 KB and that it is not the start of a new data block. Immediately following stream payload **174** is a pair comprising an identifier header **176** and identifier data **178**. The identifier header **176** includes an indication that the succeeding identifier data **178** includes the identifier for the immediately previous data block. The identifier data **178** includes the identifier that the data agent **142** generated for the data block. The data stream **171** also includes other stream header **172** and stream payload **174** pairs, which may be for SI data and/or non-SI data.

FIG. **1**H is a diagram illustrating data structures **180** that may be used to store blocks of SI data and non-SI data on a storage device (e.g., secondary storage device **108**). According to certain embodiments, data structures **180** do not form part of a native file system of the storage device. Data structures **180** include one or more volume folders **182**, one or more chunk folders **184**/**185** within the volume folder **182**, and multiple files within chunk folder **184**. Each chunk folder **184**/**185** includes a metadata file **186**/**187**, a metadata index file **188**/**189**, one or more container files **190**/**191**/**193**, and a container index file **192**/**194**. Metadata file **186**/**187** stores non-SI data blocks as well as links to SI data blocks stored in container files. Metadata index file **188**/**189** stores an index to the data in the metadata file **186**/**187**. Container files **190**/**191**/**193** store SI data blocks. Container index file **192**/**194** stores an index to container files **190**/**191**/**193**. Among other things, container index file **192**/**194** stores an indication of whether a corresponding block in a container file **190**/**191**/**193** is referred to by a link in a metadata file **186**/**187**. For example, data block B**2** in the container file **190** is referred to by a link in metadata file **187** in chunk folder **185**. Accordingly, the corresponding index entry in container index file **192** indicates that data block B**2** in container file **190** is referred to. As another example, data block B**1** in container file **191** is referred to by a link in metadata file **187**, and so the corresponding index entry in container index file **192** indicates that this data block is referred to.

As an example, data structures **180** illustrated in FIG. **1**H may have been created as a result of separate secondary copy operations involving two client computing devices **102**. For example, a first secondary copy operation on a first client computing device **102** could result in the creation of the first chunk folder **184**, and a second secondary copy operation on a second client computing device **102** could result in the creation of the second chunk folder **185**. Container files **190**/**191** in the first chunk folder **184** would contain the blocks of SI data of the first client computing device **102**. If the two client computing devices **102** have substantially similar data, the second secondary copy operation on the data of the second client computing device **102** would result in media agent **144** storing primarily links to the data blocks of the first client computing device **102** that are already stored in the container files **190**/**191**. Accordingly, while a first secondary copy operation may result in storing nearly all of the data subject to the operation, subsequent secondary storage operations involving similar data may result in substantial data storage space savings, because links to already stored data blocks can be stored instead of additional instances of data blocks.

If the operating system of the secondary storage computing device **106** on which media agent **144** operates supports sparse files, then when media agent **144** creates container files **190**/**191**/**193**, it can create them as sparse files. A sparse file is a type of file that may include empty space (e.g., a sparse file may have real data within it, such as at the beginning of the file and/or at the end of the file, but may also have empty space in it that is not storing actual data, such as a contiguous range of bytes all having a value of zero). Having container files **190**/**191**/**193** be sparse files allows media agent **144** to free up space in container files **190**/**191**/**193** when blocks of data in container files **190**/**191**/**193** no longer need to be stored on the storage devices. In some examples, media agent **144** creates a new container file **190**/**191**/**193** when a container file **190**/**191**/**193** either includes 100 blocks of data or when the size of the container file **190** exceeds 50 MB. In other examples, media agent **144** creates a new container file **190**/**191**/**193** when a container file **190**/**191**/**193** satisfies other criteria (e.g., it contains from approx. 100 to approx. 1000 blocks or when its size exceeds approximately 50 MB to 1 GB). In some cases, a file on which a secondary copy operation is performed may comprise a large number of data blocks. For example, a 100 MB file may comprise 400 data blocks of size 256 KB. If such a file is to be stored, its data blocks may span more than one container file, or even more than one chunk folder. As another example, a database file of 20 GB may comprise over 40,000 data blocks of size 512 KB. If such a database file is to be stored, its data blocks will likely span multiple container files, multiple chunk folders, and potentially multiple volume folders. Restoring such files may require accessing multiple container files, chunk folders, and/or volume folders to obtain the requisite data blocks.

Using Backup Data for Replication and Disaster Recovery ("Live Synchronization")

There is an increased demand to off-load resource intensive information management tasks (e.g., data replication tasks) away from production devices (e.g., physical or virtual client computing devices) in order to maximize production efficiency. At the same time, enterprises expect access to readily-available up-to-date recovery copies in the event of failure, with little or no production downtime.

FIG. **2**A illustrates a system **200** configured to address these and other issues by using backup or other secondary copy data to synchronize a source subsystem **201** (e.g., a production site) with a destination subsystem **203** (e.g., a failover site). Such a technique can be referred to as "live synchronization" and/or "live synchronization replication." In the illustrated embodiment, the source client computing devices **202***a* include one or more virtual machines (or "VMs") executing on one or more corresponding VM host computers **205***a*, though the source need not be virtualized. The destination site **203** may be at a location that is remote from the production site **201**, or may be located in the same data center, without limitation. One or more of the production site **201** and destination site **203** may reside at data centers at known geographic locations, or alternatively may operate "in the cloud."

The synchronization can be achieved by generally applying an ongoing stream of incremental backups from the source subsystem **201** to the destination subsystem **203**, such as according to what can be referred to as an "incremental forever" approach. FIG. **2**A illustrates an embodiment of a data flow which may be orchestrated at the direction of one or more storage managers (not shown). At step **1**, the source data agent(s) **242***a* and source media agent(s) **244***a* work together to write backup or other secondary copies of the primary data generated by the source client computing devices **202***a* into the source secondary storage device(s) **208***a*. At step **2**, the backup/secondary copies are retrieved by the source media agent(s) **244***a* from secondary storage. At step **3**, source media agent(s) **244***a* communicate the backup/secondary copies across a network to the destination media agent(s) **244***b* in destination subsystem **203**.

As shown, the data can be copied from source to destination in an incremental fashion, such that only changed blocks are transmitted, and in some cases multiple incremental backups are consolidated at the source so that only the most current changed blocks are transmitted to and applied at the destination. An example of live synchronization of virtual machines using the "incremental forever" approach is found in U.S. Patent Application No. 62/265,339 entitled "Live Synchronization and Management of Virtual Machines across Computing and Virtualization Platforms and Using Live Synchronization to Support Disaster Recovery." Moreover, a deduplicated copy can be employed to further reduce network traffic from source to destination. For instance, the system can utilize the deduplicated copy techniques described in U.S. Pat. No. 9,239,687, entitled "Systems and Methods for Retaining and Using Data Block Signatures in Data Protection Operations."

At step **4**, destination media agent(s) **244***b* write the received backup/secondary copy data to the destination secondary storage device(s) **208***b*. At step **5**, the synchronization is completed when the destination media agent(s) and destination data agent(s) **242***b* restore the backup/secondary copy data to the destination client computing device(s) **202***b*. The destination client computing device(s) **202***b* may be kept "warm" awaiting activation in case failure is detected at the source. This synchronization/replication process can incorporate the techniques described in U.S. patent application Ser. No. 14/721,971, entitled "Replication Using Deduplicated Secondary Copy Data."

Where the incremental backups are applied on a frequent, on-going basis, the synchronized copies can be viewed as mirror or replication copies. Moreover, by applying the incremental backups to the destination site **203** using backup or other secondary copy data, the production site **201** is not burdened with the synchronization operations. Because the destination site **203** can be maintained in a synchronized "warm" state, the downtime for switching over from the production site **201** to the destination site **203** is substantially less than with a typical restore from secondary storage. Thus, the production site **201** may flexibly and efficiently fail over, with minimal downtime and with relatively up-to-date data, to a destination site **203**, such as a cloud-based failover site. The destination site **203** can later be reverse synchronized back to the production site **201**, such as after repairs have been implemented or after the failure has passed.

Integrating with the Cloud Using File System Protocols

Given the ubiquity of cloud computing, it can be increasingly useful to provide data protection and other information management services in a scalable, transparent, and highly plug-able fashion. FIG. 2B illustrates an information management system **200** having an architecture that provides such advantages and incorporates use of a standard file system protocol between primary and secondary storage subsystems **217**, **218**. As shown, the use of the network file system (NFS) protocol (or any another appropriate file system protocol such as that of the Common Internet File System (CIFS)) allows data agent **242** to be moved from the primary storage subsystem **217** to the secondary storage subsystem **218**. For instance, as indicated by the dashed box **206** around data agent **242** and media agent **244**, data agent **242** can co-reside with media agent **244** on the same server (e.g., a secondary storage computing device such as component **106**), or in some other location in secondary storage subsystem **218**.

Where NFS is used, for example, secondary storage subsystem **218** allocates an NFS network path to the client computing device **202** or to one or more target applications **210** running on client computing device **202**. During a backup or other secondary copy operation, the client computing device **202** mounts the designated NFS path and writes data to that NFS path. The NFS path may be obtained from NFS path data **215** stored locally at the client computing device **202**, and which may be a copy of or otherwise derived from NFS path data **219** stored in the secondary storage subsystem **218**.

Write requests issued by client computing device(s) **202** are received by data agent **242** in secondary storage subsystem **218**, which translates the requests and works in conjunction with media agent **244** to process and write data to a secondary storage device(s) **208**, thereby creating a backup or other secondary copy. Storage manager **240** can include a pseudo-client manager **217**, which coordinates the process by, among other things, communicating information relating to client computing device **202** and application **210** (e.g., application type, client computing device identifier, etc.) to data agent **242**, obtaining appropriate NFS path data from the data agent **242** (e.g., NFS path information), and delivering such data to client computing device **202**.

Conversely, during a restore or recovery operation client computing device **202** reads from the designated NFS network path, and the read request is translated by data agent **242**. The data agent **242** then works with media agent **244** to retrieve, re-process (e.g., re-hydrate, decompress, decrypt), and forward the requested data to client computing device **202** using NFS.

By moving specialized software associated with system **200** such as data agent **242** off the client computing devices **202**, the illustrative architecture effectively decouples the client computing devices **202** from the installed components of system **200**, improving both scalability and plug-ability of system **200**. Indeed, the secondary storage subsystem **218** in such environments can be treated simply as a read/write NFS target for primary storage subsystem **217**, without the need for information management software to be installed on client computing devices **202**. As one example, an enterprise implementing a cloud production computing environment can add VM client computing devices **202** without installing and configuring specialized information management software on these VMs. Rather, backups and restores are achieved transparently, where the new VMs simply write to and read from the designated NFS path. An example of integrating with the cloud using file system protocols or so-called "infinite backup" using NFS share is found in U.S. Patent Application No. 62/294,920, entitled "Data Protection Operations Based on Network Path Information." Examples of improved data restoration scenarios based on network-path information, including using stored backups effectively as primary data sources, may be found in U.S. Patent Application No. 62/297,057, entitled "Data Restoration Operations Based on Network Path Information."

Highly Scalable Managed Data Pool Architecture

Enterprises are seeing explosive data growth in recent years, often from various applications running in geographically distributed locations. FIG. 2C shows a block diagram of an example of a highly scalable, managed data pool architecture useful in accommodating such data growth. The illustrated system **200**, which may be referred to as a "web-scale" architecture according to certain embodiments, can be readily incorporated into both open compute/storage and common-cloud architectures.

The illustrated system **200** includes a grid **245** of media agents **244** logically organized into a control tier **231** and a secondary or storage tier **233**. Media agents assigned to the storage tier **233** can be configured to manage a secondary storage pool **208** as a deduplication store and be configured to receive client write and read requests from the primary storage subsystem **217** and direct those requests to the secondary tier **233** for servicing. For instance, media agents CMA**1**-CMA**3** in the control tier **231** maintain and consult one or more deduplication databases **247**, which can include deduplication information (e.g., data block hashes, data block links, file containers for deduplicated files, etc.) sufficient to read deduplicated files from secondary storage pool **208** and write deduplicated files to secondary storage pool **208**. For instance, system **200** can incorporate any of the deduplication systems and methods shown and described in U.S. Pat. No. 9,020,900, entitled "Distributed Deduplicated Storage System," and U.S. Pat. Pub. No. 2014/0201170, entitled "High Availability Distributed Deduplicated Storage System."

Media agents SMA**1**-SMA**6** assigned to the secondary tier **233** receive write and read requests from media agents CMA**1**-CMA**3** in control tier **231**, and access secondary storage pool **208** to service those requests. Media agents CMA**1**-CMA**3** in control tier **231** can also communicate with secondary storage pool **208** and may execute read and write requests themselves (e.g., in response to requests from other control media agents CMA**1**-CMA**3**) in addition to issuing requests to media agents in secondary tier **233**. Moreover, while shown as separate from the secondary storage pool **208**, deduplication database(s) **247** can in some cases reside in storage devices in secondary storage pool **208**.

As shown, each of the media agents **244** (e.g., CMA**1**-CMA**3**, SMA**1**-SMA**6**, etc.) in grid **245** can be allocated a corresponding dedicated partition **251**A-**2511**, respectively, in secondary storage pool **208**. Each partition **251** can include a first portion **253** containing data associated with (e.g., stored by) media agent **244** corresponding to the respective partition **251**. System **200** can also implement a desired level of replication, thereby providing redundancy in the event of a failure of a media agent **244** in grid **245**. Along these lines, each partition **251** can further include a second portion **255** storing one or more replication copies of the data associated with one or more other media agents **244** in the grid.

System **200** can also be configured to allow for seamless addition of media agents **244** to grid **245** via automatic configuration. As one illustrative example, a storage manager (not shown) or other appropriate component may determine that it is appropriate to add an additional node to control tier **231**, and perform some or all of the following: (i) assess the capabilities of a newly added or otherwise available computing device as satisfying a minimum criteria to be configured as or hosting a media agent in control tier **231**; (ii) confirm that a sufficient amount of the appropriate type of storage exists to support an additional node in control tier **231** (e.g., enough disk drive capacity exists in storage pool **208** to support an additional deduplication database **247**); (iii) install appropriate media agent software on the computing device and configure the computing device according to a pre-determined template; (iv) establish a partition **251** in the storage pool **208** dedicated to the newly established media agent **244**; and (v) build any appropriate data structures (e.g., an instance of deduplication database **247**). An example of highly scalable managed data pool architecture or so-called web-scale architecture for storage and data management is found in U.S. Patent Application No. 62/273,286 entitled "Redundant and Robust Distributed Deduplication Data Storage System."

The embodiments and components thereof disclosed in FIGS. **2**A, **2**B, and **2**C, as well as those in FIGS. **1**A-**1**H, may be implemented in any combination and permutation to satisfy data storage management and information management needs at one or more locations and/or data centers.

Holistically Protecting Serverless Applications Distributed Across One or More Cloud Computing Environments

One of the benefits of the disclosed embodiments is to provide cloud-native application protection, including protection for endpoints, components, and/or assets (hereinafter "assets") associated with the application that are not already part of a traditional storage operation cell. Users receive a holistic view of the complete enterprise cloud-native application, which comprises various assets. The illustrative data protection solution is applied to the entire application entity, including its constituent assets. The illustrative embodiments orchestrate data protection for all the application's assets (e.g., file system, database, web constructs, cloud assets (e.g., Azure function, Lambda function), computing platforms (e.g., cognitive service, Azure web hook), etc. The illustrative embodiments also comprise alerting features that operate at the enterprise application level, readily identifying offline assets. Moreover, based on the holistic protection approach, an application can be restored to a "last known good state" as needed. These and other advantages and features are described in more detail in the following paragraphs and elsewhere herein.

Cloud Computing.

The National Institute of Standards and Technology (NIST) provides the following definition of Cloud Computing characteristics, service models, and deployment models:

Cloud Computing

Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model is composed of five essential characteristics, three service models, and four deployment models.

Essential Characteristics:

On-demand self-service. A consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with each service provider.

Broad network access. Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g., mobile phones, tablets, laptops, and workstations).

Resource pooling. The provider's computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand. There is a sense of location independence in that the customer generally has no control or knowledge over the exact location of the provided resources but may be able to specify location at a higher level of abstraction (e.g., country, state, or datacenter). Examples of resources include storage, processing, memory, and network bandwidth.

Rapid elasticity. Capabilities can be elastically provisioned and released, in some cases automatically, to scale rapidly outward and inward commensurate with demand. To the consumer, the capabilities available for provisioning often appear to be unlimited and can be appropriated in any quantity at any time.

Measured service. Cloud systems automatically control and optimize resource use by leveraging a metering capability[1] at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth, and active user accounts). Resource usage can be monitored, controlled, and reported, providing transparency for both the provider and consumer of the utilized service.

Service Models:

Software as a Service (SaaS). The capability provided to the consumer is to use the provider's applications running on a cloud infrastructure[2]. The applications are accessible from various client devices through either a thin client interface, such as a web browser (e.g., web-based email), or a program interface. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, storage, or even individual application capabilities, with the possible exception of limited user-specific application configuration settings.

Platform as a Service (PaaS). The capability provided to the consumer is to deploy onto the cloud infrastructure consumer-created or acquired applications created using programming languages, libraries, services, and tools supported by the provider.[3] The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, or storage, but has control over the deployed applications and possibly configuration settings for the application-hosting environment.

Infrastructure as a Service (IaaS). The capability provided to the consumer is to provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications. The consumer does not manage or control the underlying cloud infrastructure but has control over operating systems, storage, and deployed applications; and possibly limited control of select networking components (e.g., host firewalls).

Deployment Models:

Private cloud. The cloud infrastructure is provisioned for exclusive use by a single organization comprising multiple consumers (e.g., business units). It may be owned, managed, and operated by the organization, a third party, or some combination of them, and it may exist on or off premises.

Community cloud. The cloud infrastructure is provisioned for exclusive use by a specific community of consumers from organizations that have shared concerns (e.g., mission, security requirements, policy, and compliance considerations). It may be owned, managed, and operated by one or more of the organizations in the community, a third party, or some combination of them, and it may exist on or off premises.

Public cloud. The cloud infrastructure is provisioned for open use by the general public. It may be owned, managed, and operated by a business, academic, or government organization, or some combination of them. It exists on the premises of the cloud provider.

Hybrid cloud. The cloud infrastructure is a composition of two or more distinct cloud infrastructures (private, community, or public) that remain unique entities, but are bound together by standardized or proprietary technology that enables data and application portability (e.g., cloud bursting for load balancing between clouds).

Source:

Peter Mell, Timothy Grance (September 2011). The NIST Definition of Cloud Computing, National Institute of Standards and Technology: U.S. Department of Commerce. Special publication 800-145. nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf (accessed 26 Apr. 2019). Cloud computing aims to allow those who consume the services (whether individuals or organizations) to benefit from the available technologies without the need for deep knowledge about or expertise with each of them. Wikipedia, Cloud Computing, en.wikipedia.org/wiki/Cloud computing (accessed 26 Apr. 2019). "Cloud computing metaphor: the group of networked elements providing services need not be individually addressed or managed by users; instead, the entire provider-managed suite of hardware and software can be thought of as an amorphous cloud." Id.

Cloud Service Accounts and Variability in Cloud Services.

Cloud service providers such as Amazon, Microsoft, Alibaba, Google, Salesforce, Cisco, etc. provide access to their particular cloud services via cloud service accounts, such as corporate accounts, departmental accounts, individual user accounts, etc. Each cloud service account typically has authentication features, e.g., passwords, certificates, etc., to restrict and control access to the cloud service. Each account also might have service level guarantees and/or other terms and conditions between the cloud service provider and the service subscriber, e.g., a company, a government agency, an individual user. A subscribing entity might have multiple accounts with a cloud service provider, such as an account for the Engineering department, an account for the Finance department, an account for the Human Resources department, other accounts for individual company users, etc., without limitation. Each cloud service account carries different authentication, even though the services subscriber is the same entity.

Different cloud service accounts might differ not just in service level guarantees, but might include different services. For example, one account might include long-term storage resources, whereas another account might be limited to ordinary data storage. For example, some accounts might have access to data processing functions supplied by the cloud service provider, such as machine learning algorithms, statistical analysis packages, etc., whereas other accounts might lack such features. Accordingly, the resources available to the user(s) of cloud service accounts can vary as between accounts, even if the accounts have the same subscriber and the same cloud service provider. Thus, the user experience and the technologies available as between cloud service accounts can vary significantly. Thus, when considering cloud computing, the specifics of cloud service accounts can play a role in the availability and/or portability of resources. Crossing account boundaries can pose technological barriers when considering migration of applications and their cloud services assets.

Cloud Availability Zones.

"Availability zones (AZs) are isolated locations within . . . regions from which public cloud services originate and operate. Regions are geographic locations in which public cloud service providers' data centers reside. Businesses choose one or multiple worldwide availability zones for their services depending on business needs. Businesses select availability zones for a variety of reasons, including compliance and proximity to end customers. Cloud administrators can also choose to replicate services across multiple availability zones to decrease latency or protect resources. Admins can move resources to another availability zone in the event of an outage. Certain cloud services may also be limited to particular regions or AZs." Source: Margaret Rouse, Definition of Availability Zones, TechTarget, searchaws.techtarget.com/definition/availability-zones (accessed 26 Apr. 2019).

Here is a vendor-specific example of how cloud service availability zones are organized in the Google Cloud: "Certain [Google] Compute Engine resources live in regions or zones. A region is a specific geographical location where you can run your resources. Each region has one or more zones; most regions have three or more zones. For example, the us-centrall region denotes a region in the Central United States that has zones us-central1-a, us-central1-b, us-central1-c, and us-central1-f. Resources that live in a zone, such as instances or persistent disks, are referred to as zonal resources. Other resources, like static external IP addresses, are regional. Regional resources can be used by any resources in that region, regardless of zone, while zonal resources can only be used by other resources in the same zone. For example, disks and instances are both zonal resources. To attach a disk to an instance, both resources must be in the same zone. Similarly, if you want to assign a static IP address to an instance, the instance must be in the same region as the static IP. Only certain resources are region- or zone-specific. Other resources, such as images, are global resources that can be used by any other resources across any location. For information on global, regional, and zonal Compute Engine resources, see Global, Regional, and Zonal Resources." Source: Google Cloud Regions and Zones, cloud.google.com/compute/docs/regions-zones/ (accessed 26 Apr. 2019) (emphasis added).

Accordingly, when considering cloud computing, availability zones can play a role in the availability and/or portability of resources. Crossing zone boundaries can pose technological barriers when considering migration of applications and their cloud service assets, even when the different availability zones are supplied by the same cloud service provider.

Traditional Non-Cloud ("On-Premises") Data Centers are Distinguishable from Cloud Computing.

Traditional data centers generally do not have cloud computing characteristics. For example, the user experience is generally different, for example in regard to the name space(s) used for the storage, computing, and network resources. Moreover, substantial increases in resources needed by a user are not provisioned on demand. A traditional data center is physically located within the enterprise/organization that owns it. A traditional non-cloud data center might comprise computing resources such as servers, mainframes, virtual servers/clusters, etc.; and/or data storage resources, such as network-attached storage, storage area networks, tape libraries, etc. The owner of the traditional data center procures hardware, software, and network infrastructure (including making the associated capital investments); and manages going-forward planning for the data center. A traditional data center is staffed by professional Information Technology (IT) personnel, who are responsible for the data center's configuration, operation, upgrades, and maintenance. Thus, a traditional non-cloud data center can be thought of as self-managed by its owner/operator for the benefit of in-house users, as compared to cloud computing, which is managed by the cloud service provider and supplied as a service to outside subscribers. Clearly, a cloud computing service also has hardware, software, and networking infrastructure and professionals staffing it, as well as having an owner responsible for housing and paying for the infrastructure. However, the cloud computing service is consumed differently, served differently, and deployed differently compared to non-cloud data centers. Traditional non-cloud data centers are sometimes referred to as "on-premises" data centers, because their facilities are literally within the bounds of the organization that owns the data center. Cloud service providers' data centers generally are not within the bounds of the subscriber organization and are consumed "at a distance" "in the cloud."

Accordingly, when considering cloud computing versus non-cloud data center deployment, the choice can play a role in the availability and/or portability of resources. Crossing boundaries between non-cloud data centers and cloud computing can pose technological barriers. For example, storing a database at a non-cloud data center might require different resources and/or access features/controls than storing the database at a cloud computing service. Thus, moving the database from the non-cloud data center to a cloud service account may require data conversion, re-configuration, and/or adaptation that go above and beyond merely copying the database. Conversely, moving data, applications, and/or web services from cloud computing to a non-cloud data center also can involve data conversion, re-configuration, and/or adaptation to ensure success.

Service Models.

Differences in service models, comparing non-cloud "on-premises" data centers versus IaaS versus PaaS versus SaaS, can yield different performance and cost profiles. Different service models can affect resource availability and/or portability of distributed/serverless applications, at least because the management of different resources rests with different providers and governed by different terms and conditions. See, e.g., Stephen Watts, SaaS vs PaaS vs IaaS: What's The Difference and How To Choose, BMC Blogs, BMC Software, Inc., www.bmc.com/blogs/saas-vs-paas-vs-iaas-whats-the-difference-and-how-to-choose/(accessed 26 Apr. 2019).

Need For Technological Solutions to Obtain Coordinated Point-In-Time Copies and Enable Migration Of Serverless/Distributed Applications.

Consistent and coordinated point-in-time copies are needed for protecting and/or migrating distributed/serverless applications and/or data. Moving data and/or applications between non-cloud data centers and cloud computing services, between cloud computing services (e.g., from different service providers), between cloud availability zones, and even between cloud service accounts can pose technological barriers that need to be overcome. Likewise, protecting data and applications that are distributed within or span the boundaries of non-cloud data centers, cloud computing services, cloud availability zones, and/or cloud service accounts require certain technological solutions to overcome the boundary differences.

FIG. **3**A is a block diagram illustrating some salient portions of a system **300** for holistically protecting serverless applications in a cloud computing environment, according to an illustrative embodiment of the present invention. The present figure comprises three client computing devices **102**, each one accessing one or more serverless applications **310** operating in a cloud computing environment **301**. An illustrative data storage management system **300** also operates in the cloud computing environment **301**. Example cloud computing environments are provided by Amazon Web Services (AWS), Microsoft Azure, Google Cloud, Alibaba Cloud, etc., without limitation (see also Appendix A). The bold arrows from each serverless application to data storage management system **300** represent data protection operations provided by system **300** according to one or more illustrative embodiments, e.g., various kinds of backup operations, continuous replication, live synchronization, etc., without limitation.

System **300** is a data storage management system analogous to system **100** and enhanced to provide holistic protection for serverless applications distributed across one or more cloud computing environments and/or non-cloud data centers. Thus, the illustrative embodiments are not limited to cloud-based solutions. System **300** comprises certain components for data protection, such as a storage manager, a management database, data agents(s), media agent(s), and optionally secondary storage for storing copies of the subject serverless applications. The present figure depicts system **300** operating within the cloud computing environment **301**, meaning that the system's components operate within the same cloud computing environment **301** that hosts the serverless applications, but system **300** is not so limited, as described in more detail elsewhere herein. Although system **300** is shown here within the cloud computing environment **301**, it need not operate within the same cloud account as the subject application, without limitation.

FIG. **3**B is a block diagram illustrating some salient portions of system **300** for holistically protecting serverless applications from a cloud computing environment, wherein system **300** operates outside the cloud computing environment **301** that hosts the subject serverless applications **310**, according to an illustrative embodiment. Data storage management system **300** as depicted here operates outside the cloud computing environment **301** that hosts the serverless applications **310**, such as in a non-cloud data center, in another cloud availability zone of the same cloud service provider (e.g., Amazon East versus Amazon West), or in another service provider's cloud computing environment altogether (e.g., Microsoft Azure), without limitation. In some embodiments, system **300** comprises one or more components that are instantiated within the cloud computing environment **301** that hosts the subject applications, such as data agents, while other components (e.g., storage manager and/or media agents) operate outside the cloud **301**. In other embodiments, all components of system **300** are implemented outside the applications' cloud computing environment **301**.

FIG. **3**C is a block diagram illustrating some salient portions of system **300** for holistically protecting serverless applications from a plurality of cloud computing environments, wherein system **300** operates outside the cloud, according to an illustrative embodiment. The present figure depicts an illustrative multi-cloud environment in which different cloud computing environments (e.g., **301** A and **301**B) host serverless applications **310**, all of which are protected by data management system **300**. Thus, data storage management system **300** is configured to protect a plurality of serverless applications **310** from a variety of source clouds **301**, without limitation. FIG. **14** depicts another multi-cloud configuration in which a serverless application **310** operates from multiple clouds, i.e., has application assets distributed outside its home cloud computing environment—and system **300** is configured to protect such applications as well.

FIG. **3**D is a block diagram illustrating some salient portions of system **300** for holistically protecting a serverless application in a first cloud computing environment **301**A and migrating the serverless application to a second cloud computing environment **301**B, according to an illustrative embodiment. The present figure depicts data storage management system **300** configured to protect a serverless application **310** and restore it to the same source cloud computing environment **301**A. System **300** is further configured to manage migration of the serverless application **310** from one cloud computing environment (e.g., **301**A) to one or more other computing environments (e.g., **301**B), such as to other cloud availability zone(s), to another service provider's cloud computing environment, and/or to a non-cloud data center. System **300** is further configured to manage migration of a serverless application **310** from one cloud service account to another service account, though not depicted in the present figure. Migration of applications from non-cloud data centers to cloud computing is also supported by system **300** in certain embodiments, though not depicted in the present figure.

FIG. **4** is a block diagram illustrating some salient assets of a serverless application **310** that is protected by data storage management system **300**, according to an illustrative embodiment. The present figure depicts a client computing device **102** accessing a serverless application **310**, which is protected by data storage management system **300**, i.e., backed up using one or more storage management operations. The serverless application **310** comprises assets, i.e., resources that it uses in whole or in part to perform its functionality. The depicted serverless application comprises the following illustrative assets: a gateway **421**; data storage **422**; computing resources **423**; databases **424**; native cloud applications such as web sites and others **425**; virtual machines **426**; etc. without limitations. More cloud computing assets are presented and discussed in FIG. **12**. Notably, the assets depicted here are logically part of the serverless application **310**, but they are not necessarily in the same cloud computing environment. For example, one or more databases could be in another computing environment, whether cloud or non-cloud; likewise, data storage might be in yet another computing environment, whether cloud or non-cloud. As is shown herein, data storage management system **300** discovers the application's assets regardless of where they reside/operate and treats them collectively as an application entity. See also FIGS. **13** and **14** for more details on an exemplary application service definition associated with an application **310**.

Gateway function **421** provides client computing devices with API-based access to the application **310**. Gateway functions are specific to the cloud computing environment (e.g., **301**) hosting application **310**, as well as being specific to the application itself, e.g., including access controls and permissions. Although each gateway function **421** is specific to its protected target, technologies for gateways **421** are well known in the art.

Data storage **422** depicts one or more data storage resources associated with application **310**, including high-speed "local" and temporary storage, as well as slower lower-cost storage. Data storage resource technologies, whether cloud-based or non-cloud, are well known in the art.

Computing resources **423** depict any number of data processing resources available for application **310** to consume. For example, Amazon AWS provides so-called "Lambda functions" defined as "anonymous computing functions not bound to an identity" and made available as containerized instances that are executed when an event is triggered. See en.wikipedia.orq/wiki/AWS Lambda (accessed Oct. 17, 2019). "Each such execution is run in a new environment so access to the execution context of previous and subsequent runs is not possible." Id. Other cloud service providers have their own suites of cloud computing services are resources. An application designer chooses features from pre-existing computing resources **423** to use by application **310** as needed. Computing resources **423** are well known in the art.

Databases **424** depict any number of database data repositories with or without an associated database management system. Again, an application designer chooses data and/or features from databases **424** to use by application **310** as needed. For example, a census database and an associated database management system for searching and manipulating data therein. Proprietary and open source DBMSs are available as cloud-based databases-as-a-service (DBaaS). Typically, a customer activates one or more instances of one or more desired DBaaS (e.g., MySQL, Oracle, PostgreSQL, Microsoft SQL, Aurora, Maria DB, Dynamo DB, Redshift, Data Lake, Microsoft Office 365 Suite, Cosmos DB, MongoDB, IBM DB2, etc.) in a cloud computing account, thus obtaining immediate access to a fully featured DBMS. The term "DBMS" is used herein to identify the kind of database management system (e.g., MySQL, Oracle, DB2, etc.), whereas the term "DBaaS" is used herein to refer to a DBMS served as a service in a cloud computing environment. Accordingly, a MySQL DBMS might execute as a first DBaaS instance, a PostgreSQL DBMS might execute as a second DBaaS instance, and a second PostgreSQL DBMS (the same or different version as the first) might execute as a third DBaaS instance. And so on.

Native cloud applications **425** depict any number of readily available applications such as web sites, web service, and others that are well known in the art. Again, an application designer chooses data and/or features from native cloud applications **425** to use by application **310** as needed.

Virtual machines **426** represent virtualized computing resources available to application **310**, such as for executing the application's custom code that aggregates the application's functionality (e.g., providing weather forecasts to consumers). Virtualized computing resources can be spun up and spun down on demand, enabling applications **310** to scale-out with demand. Virtual machines **426** are well known on the art.

Thus, although the kinds of assets described herein are well known in the art, the architecture of each application **310** is defined by its designer/implementers and likewise application **310**'s functionality. Data storage management system **300** is designed to intelligently and flexibly capture and protect each serverless application **310**, regardless of its unique architecture and feature functionality, as described in further detail herein.

FIG. **5**A is a block diagram illustrating some salient portions of system **300** protecting a serverless application **310** comprising a plurality of assets, according to an illustrative embodiment. As shown in FIG. **4**, the serverless application **310** comprises assets that are logically part of or associated with the application, but do not necessarily reside and/or operate in the same cloud computing environment. Data storage management system **300** illustratively comprises: storage manager **540**, management database **546**, data agents **542**, media agents **544**, and any number of point-in-time copies **516** of the application entity (i.e., sets of copies of application assets). The data storage resources for storing the point-in-time copies **516** are not necessarily part of system **300**, such as cloud-based storage that is accessible to but not included in system **300**. Any kind of suitable data storage devices, systems, configurations, and/or resources can be implemented to store point-in-time copies **516**. Moreover, system **300** is further configured to apply additional storage operations to copies **516**, such as making auxiliary copies, making archive copies, replicating copies **516**, live synchronizing copies **516**, etc., without limitation.

Data agents **524**. The present figure depicts data agents **542**, which are analogous to data agents **142** and comprise additional features for operating in system **300**. For example, different types of data agent **542** are invoked for protecting different kinds of application assets. For example, a data agent specialized for Oracle databases might be invoked for protecting an Oracle database, whereas a data agent for protecting files and file systems might be invoked for protecting the application configuration file(s) of a native cloud application.

Media agents **544** are invoked for generating secondary copies **516** and storing them to suitable secondary storage within or outside the cloud computing environment that hosts application **310**. Media agents **544** are analogous to media agents **144** and additionally comprise features for receiving, coordinating, indexing, and tracking sets of copies **516** that correspond to a certain point-in-time view of application **310**.

Storage manager **540** is generally responsible for managing system **300**, including managing storage management operations, invoking and instructing components such as data agents **542** and media agents **544**, and tracking secondary copies **516**. Storage manager **540** is analogous to storage manager **140** and, as shown in the next figure, storage manager **540** also has certain features specifically designed for protecting, restoring, and/or migrating serverless distributed applications.

It should be noted that, although data agents **542** and media agents **544** are components of data storage management system **300**, they might operate in distinct computing environments. For example, in some embodiments, storage manager **540** invokes data agents **542** in the same cloud computing environment as the target assets, generally for performance reasons; and storage manager **540** invokes media agents **544** in a different computing environment, e.g., in a non-cloud data center where the secondary storage is located—and also for performance reasons. Thus, although data agents, media agents, management database, and storage manager are all components of a system **300**, they need not operate in the same computing environment nor do they have to co-operate or co-reside with the target data sources and/or data destinations. The architecture of system **300** provides for flexibility, scalability, and configurability to meet the needs for each serverless/distributed application scenario. Although much of the discussion herein invokes cloud computing terminology, certain embodiments operate within and between traditional non-cloud data centers, e.g., supporting migration of distributed applications between non-cloud data centers.

FIG. **5**B is a block diagram illustrating some salient portions of system **300**, including storage manager **540** and management database **546**, according to an illustrative embodiment.

Storage manager **540** is specially configured with features for collecting information about application assets and for orchestrating backup, restore, and/or migration of serverless applications. The former feature is illustratively implemented as an application collection layer **560**. The latter feature is illustratively implemented as an application entity orchestration layer **570**. Layer **560** and layer **570** are shown here as distinct functional modules to ease the reader's understanding of the present disclosure, but the invention is not so limited. In some embodiments, one or more of these features are implemented in one or more other components, such as a proxy server, a media agent, a data agent, etc., without limitation. For example, and without limitation, a proxy server that comprises data agents and media agents also could comprise application collection layer **560**.

Management database **546** comprises a number of new and/or changed data structures needed for system **300**, including without limitation: an application entity definition **552**; an application asset mapping **554**; preferences **556** applicable to the application entity such as policies for backups, replication, live synchronization, Recovery Point Objective (RPO), etc.; and indexing information **558** that associates and tracks sets of copies **516** that are associated with a particular point-in-time view of serverless application **310**. These are depicted here as distinct data structures to ease the reader's understanding of the present disclosure, but the invention is not so limited. Moreover, in some embodiments, some or all of these data structures are also stored in other system components, such as indexing information retained in media agents, etc.

FIG. **6** depicts some salient operations of a method **600** according to an illustrative embodiment. Method **600** is generally performed by one or more components of data storage management system **300**. System **300** is configured to:

- Discover information about a serverless distributed application, including acquiring credentials to access the application's cloud service account and discovering application assets and asset dependencies (block **602** and FIG. **7**A);
- Define a corresponding application entity in the data storage management system (block **602** and FIG. **7**B);
- Define storage management preferences (e.g., policies) for the application entity as a whole, which apply to the application's assets collectively. Preference examples include storage policies, schedule policies, retention policies, destinations for copies, deduplication preferences, recovery point objective (RPO), etc. (block **604**);
- Perform storage management operations upon the application entity, resulting in a point-in-time set of copies of application assets. Operations include one or more backups, replication, live synchronization, etc., without limitation (block **606** and FIGS. **8**A, **8**B, and **9**);

- Restore the application to the source cloud computing environment using a point-in-time set of copies of application assets (block **608**); and
- Migrate the application to another computing environment (different from the source) using a point-in-time set of copies of application assets and apply conversion as needed to re-activate the application in its new environment (block **610** and FIG. **10**).

More details regarding these features are given in other figures. Block **608** is similar to block **610**, but instead of migrating data to a destination different from the source, it performs some of the same operation in reference to the same computing environment of the source application **310**. A person having ordinary skill in the art will know how to perform block **608** after reading and understanding the present disclosure.

FIG. **7**A depicts some salient operations of block **602** in method **600**. Block **602** is directed to discovering information about a "serverless" distributed application **310** and defining a corresponding application entity in data storage management system **300**. The operations of block **602** are illustratively performed by storage manager **540**, e.g., using application collection layer **560**, but the invention is not so limited. The operations of block **602** continue in FIG. **7**B.

At block **702**, method **600** exposes a cloud service account to storage manager **540**. This operation may be a "push" performed from the cloud service account configured with a backup destination. In some embodiments, this operation is a "pull" by storage manager **540** becoming aware of a new protection target. In some embodiments, URLs and access credentials are administered into storage manager **540** by administrators before storage manager **540** can gain access to the cloud service account.

At block **704**, method **600** selects an application **310** from the cloud service account's catalog. Again, this operation may be a "push" performed from the cloud service account configured to communicate with storage manager **540**. In some embodiments, this operation is a "pull" by storage manager **540** exploring the cloud service account.

At block **706**, storage manager **540** gains access to the application's cloud services definition in the cloud service account. Storage manager **540** comprises a number of features for exploring cloud service accounts to find cloud services definitions, e.g., knowing how and where to find cloud services definitions (or the equivalent thereof) in different cloud service accounts.

At block **708**, method **600**, e.g., using application collection layer **560**, discovers the application's cloud assets, e.g., based on a cloud services definition in the cloud service account. When an express cloud services definition is not readily discernible, application collection layer **560** is configured to explore information associated with application **310** in the cloud service account and to infer what its cloud assets might be.

At block **710**, method **600**, e.g., using application collection layer **560**, determines dependencies among the application's assets, e.g., based on the cloud services definition in the cloud service account. When an express cloud services definition is not readily discernible, application collection layer **560** is configured to explore information associated with application **310** in the cloud service account and to infer what the dependencies might be. For example, layer **560** determines that a data processing function F depends upon (a) user-input configuration parameters C and (b) additional data extracted from a database D. Accordingly, layer **560** determines that function F depends on assets C and D.

At block **712**, blocks **702-710** repeat if the application is distributed across other cloud service accounts, cloud accessibility zones, cloud services, and/or non-cloud data center environments. Control then passes to block **714** in FIG. **7**B.

FIG. **7**B depicts some additional salient operations of block **602** in method **600**, continuing from FIG. **7**A.

At block **714** of method **600** storage manager **540** creates an application entity in the data storage management system **300** having a unique entity identifier, e.g., weather_app_2019. Illustratively the definition is stored in management database **546**.

At block **716**, storage manager **540** generates an application entity definition **552** comprising the discovered cloud assets, configurations, properties, permissions, governance, regulatory restrictions, cost considerations, etc. Generally, the entity definition **552** comprises parameters and information to know about corresponding serverless application **310**. At least some of the contents of definition **552** are used by system **300** to protect, restore, and/or migrate application **310**.

At block **718**, an asset mapping **554** is created for the application that defines dependencies among the cloud assets of the application. See also block **710**. Illustratively, application collection layer **560** generates the application asset mapping **554** that captures dependencies gleaned during the asset discovery process. These relationships are stored in the asset mapping **554** and they may affect an "order of operations," which is used when the storage management operation is initiated. For example, the order of operations might capture the user-input configuration parameters C first, followed by capturing the database D, followed by capturing the data processing function's executable files (e.g., binaries, etc.) in that order. In some embodiments, the order of operations is stored within or in association with the asset mapping. In other embodiments, the order of operations is determined on demand, when the storage management operation is initiated. In configurations where layer **560** executes at a component apart from storage manager **540**, e.g., at a media agent **544**, at a data agent **542**, the component transmits asset mapping **552** to storage manager **540**, which stores it to management database **546**.

At block **720**, storage manager **540** receives and/or generates preferences **556** for the application entity, e.g., backup, replication, destination, RPO, etc. These are provided by an administrator, downloaded from a template, and/or defaulted into storage manager **540** and are stored in management database **546**.

At block **722**, storage manager **540** stores application entity definition **552**, asset mapping **554**, preferences **556**, and any other ancillary data structures to management database **546**.

FIG. **8**A depicts some salient operations of block **606** in method **600**. Block **606** is generally directed to performing storage management operations upon the application entity, resulting in a point-in-time set of copies of application assets. The operations of block **606** are illustratively performed by one or more of the components of system **300**, including storage manager **540**, data agents **542**, and media agents **544**, resulting in a coordinated (point-in-time) set of secondary copies **516**. The operations of block **606** continue in FIG. **8**B and FIG. **9** (block **808**).

At block **802**, based on a storage management preference **556** for the application entity, an appropriate storage management operation is initiated, e.g., backup, replication, live synchronization, etc. Illustratively, storage manager **540** initiates the operation, which may be based on timing or criteria that have been satisfied to trigger the operation.

At block **804**, storage manager **540** identifies assets for the application entity, e.g., based on the application entity definition **552** and/or asset mapping **554** in the management database **546**. One or more assets **421-426** etc. are identified here.

At block **806**, based on the dependencies in the application asset mapping **554**, storage manager **540** defines an order of operations, as described in block **718**.

At block **808**, for each application asset in the application asset mapping **554** and using the order of operations determined at block **802**, the storage management operation is performed as instructed, resulting in a point-in-time set of copies **516** of application assets. More details are given in FIG. **9**. Control passes to block **820** in FIG. **8**B.

FIG. **8**B depicts some additional salient operations of block **606** in method **600** continued from FIG. **8**A.

At block **820**, media agent **544** stores the point-in-time set of copies **516** to secondary storage according to the preference that triggered the storage management operation, e.g., same-cloud storage, different-cloud storage, and/or data center, etc.

At block **822**, media agent **544** generates and indexes associations among the application **310**, the time of the storage management operation (the point-in-time), and the asset copies **116** in the point-in-time set **516**. This step aggregates the distinct operations into a coordinated set **516** that represents the point-in0time view of application **310**.

At block **824**, method **600** stores the information (e.g., associations and indexing information) to the management database **546**, media agent(s) **544**, and/or secondary storage. In some embodiments, the stored information is distributed among distinct components, e.g., pointers to media agent **544** stored at management database **546**, pointers to the set **516** stored at media agent **544**, etc., without limitation.

At block **826**, method **600** reports performance statistics collected during the operation with respect to application **310**, e.g., recommending changes to protection schedules to better align with application protection. In some embodiments, the data storage management system **300** (e.g., storage manager **540**) recommends an improved and coordinated schedule for protecting the various assets that are associated with application **310**. For example, if a certain database is associated with application **310**, it will be backed up according to the preferences for the application-entity. But it might also be protected according to preferences for the database entity itself, thus creating separate rounds of backup operations for the same database. The storage manager **540** will generate recommendations for a coordinated schedule that unifies the database backups and reduces redundancies. The coordinated schedule will be recommended to system administrators for approval. In alternative embodiments, the coordinated schedule will be automatically implemented when the system detects a redundancy between application-entity protection and individual asset protection, without limitation.

At this point, method **600** ends and the present storage management operation is complete in regard to application **310**.

FIG. **9** depicts some salient operations of block **808** in block **606** of method **600**. Block **808** is generally directed to, for each application asset in the application asset mapping, and using the order of operations, performing the storage management operation as instructed. Notably, the storage management operation may comprise a number of distinct operations, one or more operations targeting each individual asset. Coordination among them is performed by application entity orchestration layer **570**. For example, and in reference to FIG. **5**A, storage manager **540**, e.g., using application entity orchestration layer **570**, instructs each data agent **542** associated with an asset of application **310** to quiesce the asset, if applicable, take a snapshot, if applicable, process the source data from the respective asset, and transmit the processed source data to the media agent **544** assigned to the operation. Storage manager **540**, e.g., using application entity orchestration layer **570**, instructs media agent **544** to further process the data received from the data agents into a set of secondary copies **516**. The set **516** comprises one or more secondary copies **116** created from each of the assets of application **310**.

At block **902**, based on the order of operations, storage manager **540** invokes and instructs a first data agent **542** and one or more first media agents **544** to generate first copies of a first application asset, e.g., **422**. In some cloud computing environments, data agents **542** are kept dormant in between storage operations and are invoked on demand as needed. Likewise in regard to media agents **544**. This is made possible by using virtualized resources available in the cloud computing environment to run resources only as needed. Alternatively, data agents **542** and media agents **544** are always on, which is typical in non-cloud data centers.

At block **904**, further based on the order of operations, storage manager **540** invokes and instructs a next data agent **542** and one or more media agents **544** to generate corresponding copies of a next application asset, e.g., **423**, sequentially or in parallel, until the order of operation has been traversed.

At block **906**, the point-in-time set **516** of copies of application assets, which are mutually consistent in time, is generated.

FIG. **10** depicts some salient operations of block **610** in method **600**. Block **610** is generally directed to migrating the application to a computing environment (cloud, or non-cloud) different from the application source environment using a point-in-time set **516** of copies of the application assets that were generated and stored at block **606**. The operations of block **610** are illustratively performed by one or more of the components of system **300**, including storage manager **540**, data agents **542**, and media agents **544**.

At block **1002**, method **600** exposes a cloud service account to storage manager **540**, which is in a different cloud computing environment from the source application **310**. This block is similar to block **702**.

At block **1004**, method **600** identifies a point-in-time set of copies of the application assets. This operation may be based on input received by storage manager **540** from a user seeking to migrate a certain point-in-time backup (e.g., dated Monday Oct. 14, 2019) of an application entity (e.g., weather_app_2019) to a new destination.

At block **1006**, based on the dependencies in the application asset mapping **554** for the application entity, the storage manager **540** defines an order of operations. See also block **718**, **806**. The order of operations may differ from an order of operation determined for the data protection operation at blocks **718**, **806**.

At block **1008**, based on the order of operations, storage manager **540** instructs a first data agent **542** and first media agent(s) **544** to migrate, from a set **516**, first copies **116** of a first application asset to the destination cloud service account and/or non-cloud data center. This may include data conversion, if applicable, in cases where simply restoring the copy **116** is not enough to make it operational at the destination. For example, configuration parameters may have to be translated from source to destination. Sizing parameters may have to be adjusted, etc.

At block **1010**, further based on the order of operations, storage manager **540** instructs a next data agent **542** and media agent(s) **544** to migrate corresponding copies **116** of a next application asset, until the order of operations is traversed. Thus, other copies **116** from set **516** are restored, which again, may require conversion as needed.

At block **1012**, method **600** generates a suitable cloud services definition for the restored application **310** comprising the restored application assets. This step is used when the destination is a cloud computing environment but may be applied for non-cloud data centers as well. In some embodiments, conversion of configuration parameters may be needed from one cloud computing environment to another. Storage manager **540**, e.g., using application entity orchestration layer **570**, is configured to analyze source and destination in a migration operation, determine differences, determine suitable conversions, and apply the conversions as needed. At this point, the application migration is complete and the point-in-time version of the application **310** available at the destination, bringing method **600** to an end.

FIG. **11** depicts an illustrative order of operations **1101** that might be employed in a storage management operation such as a full backup, an incremental backup, a replication, a live synchronization, etc. The depicted order is not limiting. The order of operations is based on dependencies among application assets, e.g., captured in asset mapping **554**. The order of operations need not be linear or single-threaded and allows for parallel operations when assets are not dependent. As noted in FIG. **10**, an order of operations in a restore or migration operation may differ from the order of operations used in the original protection (e.g., backup etc.) operation of FIG. **7A/7B**. Hence, preferably, the order of operations **1101** is determined on demand at the time the storage management operation is initiated, though the invention is not so limited.

FIG. **12** depicts an illustrative list **1201** (catalog, inventory) of cloud assets of an application **310** configured in a cloud account. This list in included here for illustrative purposes and is neither limiting nor exclusive. Different assets are used for different serverless applications, and they may carry different names or designations depending on the cloud service provider's nomenclature. Some examples of application assets are shown in FIG. **4**.

FIG. **13** depicts an illustrative serverless distributed application service definition in a cloud account. This figure depicts: a client computing device **102** in communication with a cloud computing gateway **421**. The arrows depict illustrative logical data paths among the application assets. Accordingly, data input/triggers from the gateway **421** are transmitted to first computing resources **423**A (e.g., a lambda function) that applies a first round of data processing; the results pass to second computing resources **423**B that additionally obtain data from an algorithms database **424** and produce a second round of "final" results, which are stored to temporary data storage **1308**. The final results are then stored to persistent data storage **1309**. The final results are transmitted back to the gateway **421**, which takes care of transmitting them to the client computing device **102**, thus completing an execution cycle of the serverless application **310**.

This diagram not only depicts the functions and logical data flows, but also provides a basis for determining dependencies. Thus, as depicted here and without limitation, computing resources **423** depend from computing resources **423** A and from database **424**, whereas output **1308** depends from computing resources **423**B. Thus, as depicted here and without limitation, an illustrative order of operations for data protection operations would be: (i) database **424**; (ii) computing resources **423**A; (iii) computing resources **423**B; (iv) output **1308**; and (v) output **1309**, wherein (i) and (ii) can be performed in parallel. In a restore or migration operation, an illustrative restore/migration order of operations would be: (v), (iv), (iii), (ii), and (i), wherein (i) and (ii) can be performed in parallel. Depending on different cloud computing environments one or more parameters from gateway function **421** also are backed up, and then restored, converted, and/or migrated as appropriate. The ordering of handling the one or more parameters from gateway function **421** relative to handling the other application assets depends on how the source and destination environments are configured.

FIG. **14** depicts an illustrative service definition of a serverless application **310** that is distributed across multiple computing environments. This figure depicts the same assets shown in FIG. **13**, and additionally shows that the assets reside in different computing environments, i.e., this figure depicts a multi-cloud serverless application. Illustratively, the gateway **421**, the computing resources **423**A and **423**B, and the temporary data storage **1308** reside in cloud computing environment **301**A; the algorithms database **424** resides in cloud **301**B (e.g., cloud storage); and the persistent storage **1309** for final results resides in cloud C and/or a non-cloud data center **1401** distinct from clouds **301**A and **301**B. This depiction is shown here to ease the reader's understanding of the present disclosure, and any number of combinations and permutations are possible in other embodiments. System **300** is configured to protect, restore, and migrate a serverless application **310** in this or any other distributed configuration, by deploying its components appropriately as described in more detail elsewhere herein.

In regard to the figures described herein, other embodiments are possible within the scope of the present invention, such that the above-recited components, steps, blocks, operations, messages, requests, queries, and/or instructions are differently arranged, sequenced, sub-divided, organized, and/or combined. In some embodiments, a different component may initiate or execute a given operation.

Example Embodiments

Some example enumerated embodiments of the present invention are recited in this section in the form of methods, systems, and/or non-transitory computer-readable media, without limitation.

According to an illustrative embodiment, a method for protecting a serverless application operating in a cloud computing environment comprises: by a first computing device (e.g., storage manager), generating a list of assets associated with a first application operating in a first cloud computing environment accessed by a first cloud service account, wherein the first computing device comprises one or more hardware processors; by the first computing device, generating a first application-entity definition that corresponds to the first application and comprises the list of assets associated with the first application; by the first computing device, initiating a storage management operation for the first application-entity, wherein the storage management operation generates a first set of copies of the assets associated with the first application; and wherein the first set of copies represent a point-in-time copy of the first application at a time when the storage management operation was initiated for the first application-entity. The above-recited method, wherein the list of assets associated with the first application are discovered by the first computing device after accessing the cloud service account. The above-recited method, wherein the list of assets associated with the first application are discovered by the first computing device based on a cloud services definition in the cloud service account. The above-recited method further comprising: by the first computing device, based on determining one or more dependencies among the assets associated with the first application, generating an asset mapping for the first application-entity that comprises the one or more dependencies. The above-recited method wherein the first set of copies are generated according to a first order of operations based on the one or more dependencies among the assets associated with the first application. The above-recited method, wherein the first application-entity definition is stored in a database associated with the first computing device. The above-recited method further comprising: storing in the database one or more preferences for protecting the first application-entity. The above-recited method, wherein the initiating of the storage management operation for the first application-entity is based on one or more preferences associated with the first application-entity. The above-recited method, wherein the storage management operation is initiated based on a preference associated with the first application-entity. The above-recited method further comprising: based on the definition of the first application-entity, determining by the first computing device the assets associated with the first application.

The above-recited method further comprising: by the first computing device, invoking a first data agent suitable for protecting a first asset associated with the first application; by the first computing device, invoking a media agent to receive data from the first data agent, generate a first copy of the first asset, and store the first copy of the first asset to a storage device; by the first computing device, invoking a second data agent that is suitable for protecting a second asset associated with the first application; by the first computing device, invoking a media agent to receive data from the second data agent, generate a second copy of the second asset, and store the second copy to a storage device; and by the first computing device, generating an association among (a) the first application-entity, (b) a time when the storage management operation for the first application-entity was initiated, and (c) a set of copies of the assets associated with the first application, including the first copy and the second copy. The above-recited method further comprising: by the first computing device, invoking a first data agent suitable for protecting a first asset associated with the first application, wherein the first asset is selected according to an order of operations; by the first computing device, invoking a media agent to receive data from the first data agent, generate a first copy of the first asset, and store the first copy of the first asset to a storage device; by the first computing device, invoking a second data agent that is suitable for protecting a second asset associated with the first application, wherein the second asset is selected according to the order of operations; by the first computing device, invoking a media agent to receive data from the second data agent, generate a second copy of the second asset, and store the second copy to a storage device; and by the first computing device, generating an association among (a) the first application-entity, (b) a time when the storage management operation for the first application-entity was initiated, and (c) a set of copies of the assets associated with the first application, including the first copy and the second copy. The above-recited method, wherein the storage management operation comprises at least one of: a full backup of the first application-entity; an incremental backup of the first application-entity; and a differential backup of the first application-entity. The above-recited method, wherein the storage management operation comprises at least one of: continuous replication of the first application-entity; and live synchronization of the first application-entity. The above-recited method, wherein the assets associated with the first application comprise at least one of: a virtual computing environment, a template for the virtual computing environment, secure login information, storage for temporary data, persistent storage, metadata, virtual network resources and/or configurations, a snapshot, a security group, a configuration file, a lambda function, a native cloud application, a virtual machine, a database, a file storage resource, a block storage resource.

Recommend a Coordinated Backup Schedule.

The above-recited method, wherein the first computing device is configured to generate a coordinated schedule for backing up an asset of the first application, wherein the asset is configured as a distinct asset-entity in the data storage management system with preferences for protecting the asset-entity that are distinct from preferences for protecting the application-entity, and wherein the coordinated schedule coordinates timing of storage management operations for the application-entity and for the asset-entity such that copies of the asset-entity are generated concurrently with copies of the application-entity. The above-recited method, wherein the first computing device is configured to generate a coordinated schedule for backing up a database that is an asset of the first application, wherein the database is configured as a distinct entity in the data storage management system with preferences for protecting the database that are distinct from preferences for protecting the application-entity, and wherein the coordinated schedule coordinates timing of storage management operations for the application-entity and for the database such that copies of the database are generated concurrently with copies of the application-entity.

Computing Device Alternatives.

The above-recited method, wherein the first computing device operates in a non-cloud data center that is distinct from the first cloud computing environment of the first application. The above-recited method, wherein the first computing device comprises a virtual computing resource that executes in the first cloud computing environment. The above-recited method, wherein the first computing device comprises a virtual computing resource that executes in a second cloud computing environment different from the first cloud computing environment. The above-recited method, wherein the first computing device comprises a virtual computing resource that executes in a different availability zone from the first cloud computing environment. The above-recited method, wherein the first computing device comprises a virtual computing resource that executes in a different cloud service account from the first cloud service account in the first cloud computing environment.

Data Storage Alternatives.

The above-recited method, wherein the first set of copies of the assets associated with the first application are stored in storage resources in the first cloud computing environment. The above-recited method, wherein the first set of copies of the one or more cloud assets associated with the first application are stored in storage resources in a second cloud computing environment different from the first cloud computing environment. The above-recited method, wherein the first set of copies of the one or more cloud assets associated with the first application are stored in a different availability zone from the first cloud computing environment. The above-recited method, wherein the first set of copies of the one or more cloud assets associated with the first application are stored to storage resources in a non-cloud data center that is distinct from the first cloud computing environment. The above-recited method, wherein the first set of copies of the one or more cloud assets associated with the first application are stored in storage resources accessible by a different cloud service account from the first cloud service account.

Restore the App to the Same Cloud Account.

The above-recited method further comprising: causing by the first computing device the first set of copies to be restored to the first cloud computing environment; and activating the first application at the first cloud service account, wherein the first application uses assets restored from first set of copies.

Migrate the App to Another Cloud Account.

The above-recited method further comprising: causing by the first computing device the first set of copies to be migrated to a second cloud computing environment, based at least in part on generating a cloud service definition in the second cloud computing environment from at least one of: (a) the application-entity; and activating the first application in the second cloud computing environment, wherein the first application uses assets migrated from first set of copies.

Two Applications with Overlapping Assets.

The above-recited method further comprising: by the first computing device, discovering assets associated with a second application operating in the first cloud computing environment, wherein at least one asset associated with the first application is also associated with the second application; by the first computing device, generating a list of assets associated with the second application; by the first computing device, generating a second application-entity definition that corresponds to the second application and comprises the list of assets associated with the second application; by the first computing device, initiating a second storage management operation for the second application-entity, based on a set of preferences for the second application-entity, and resulting in a second set of copies of the assets associated with the second application including at least one copy of the at least one asset associated with the first application is also associated with the second application; and wherein the second set of copies represent a point-in-time copy of the second application at a time when the second storage management operation was initiated for the second application-entity.

Multi-Cloud App.

According to another exemplary embodiments, a method for protecting a serverless application operating in multiple cloud computing environments comprises: by a first computing device, generating a list of first assets associated with a first application operating in a first cloud computing environment accessed by a first cloud service account, wherein the first computing device comprises one or more hardware processors; by the first computing device, generating a first application-entity definition that corresponds to the first application and is associated with the list of first assets associated with the first application; by the first computing device, discovering in a second cloud computing environment second assets associated with the first application; by the first computing device, associating the first application-entity with the second cloud assets associated with the first application. The above-recited method further comprising: initiating by the first computing device a storage management operation for the first application-entity, wherein the storage management operation generates a first set of copies of the first assets and of the second assets associated with the first application; and wherein the first set of copies represent a point-in-time copy of the first application at a time when the storage management operation was initiated for the first application-entity.

Restore the Multi-Cloud App to the Same Cloud Accounts.

The above-recited method further comprising: causing by the first computing device the first set of copies to be restored, wherein some of the first set of copies corresponding to the first assets are be restored to the first cloud computing environment, and further wherein some of the first set of copies corresponding to the second assets are restored to the second cloud computing environment; and activating the first application at the first cloud service account, wherein the first application uses the restored first cloud assets in the first cloud computing environment and the restored second assets in the second cloud computing environment.

Migrate the Multi-Cloud App to Another Cloud Account.

The above-recited method further comprising: causing by the first computing device the first set of copies to be migrated to a third cloud computing environment that is distinct from the first and the second cloud computing environments; and activating the first application at the third cloud computing environment, wherein the first application uses third cloud assets corresponding to the first and the second assets based on a cloud services definition that is suitable for the third cloud computing environment. The above-recited method, wherein migrating the first set of copies to the third computing environment comprises converting an asset mapping associated with the first application-entity into the cloud services definition that is suitable for the third cloud computing environment.

Migrate the Multi-Cloud App to a Non-Cloud Data Center Instead of Cloud.

The above-recited method further comprising: causing by the first computing device the first set of copies to be migrated to a non-cloud data center that is distinct from the first and the second cloud computing environments; and activating the first application at the non-cloud data center, wherein the first application uses third assets configured from the first and the second assets to configurations suitable to the non-cloud data center. The above-recited method, wherein migrating the first set of copies to the non-cloud data center comprises converting an asset mapping associated with the first application entity into the configurations suitable to the non-cloud data center.

Migrate an App from Non-Cloud Data Center to Cloud.

According to another illustrative embodiments, a method for migrating an application from a non-cloud data center to a cloud computing environment comprises: by a first computing device (e.g., storage manager), discovering first assets associated with a first application that executes in a non-cloud data center; by the first computing device, generating a first application-entity definition that corresponds to the first application and is associated with a list of the first assets; initiating by the first computing device a storage management operation for the first application-entity, wherein the storage management operation generates a first set of copies of the first assets associated with the first application; by the first computing device, accessing a first cloud computing environment; by the first computing device, generating a cloud services definition for the first application in the first cloud computing environment, based at least in part on converting an asset mapping associated with the first application-entity into the cloud services definition that is suitable for the first cloud computing environment; causing by the first computing device the first set of copies to be migrated to the first cloud computing environment; activating the first application at the first cloud computing environment, based at least in part on the cloud services definition, wherein the first application uses assets migrated from the first set of copies.

In other embodiments according to the present invention, a system or systems operates according to one or more of the methods and/or computer-readable media recited in the preceding paragraphs. In yet other embodiments, a method or methods operates according to one or more of the systems and/or computer-readable media recited in the preceding paragraphs. In yet more embodiments, a non-transitory computer-readable medium or media causes one or more computing devices having one or more processors and computer-readable memory to operate according to one or more of the systems and/or methods recited in the preceding paragraphs.

Terminology

Conditional language, such as, among others, "can," "could," "might," or "may," unless specifically stated otherwise, or otherwise understood within the context as used, is generally intended to convey that certain embodiments include, while other embodiments do not include, certain features, elements and/or steps. Thus, such conditional language is not generally intended to imply that features, elements and/or steps are in any way required for one or more embodiments or that one or more embodiments necessarily include logic for deciding, with or without user input or prompting, whether these features, elements and/or steps are included or are to be performed in any particular embodiment.

Unless the context clearly requires otherwise, throughout the description and the claims, the words "comprise," "comprising," and the like are to be construed in an inclusive sense, as opposed to an exclusive or exhaustive sense, i.e., in the sense of "including, but not limited to." As used herein, the terms "connected," "coupled," or any variant thereof means any connection or coupling, either direct or indirect, between two or more elements; the coupling or connection between the elements can be physical, logical, or a combination thereof. Additionally, the words "herein," "above," "below," and words of similar import, when used in this application, refer to this application as a whole and not to any particular portions of this application. Where the context permits, words using the singular or plural number may also include the plural or singular number respectively. The word "or" in reference to a list of two or more items, covers all of the following interpretations of the word: any one of the items in the list, all of the items in the list, and any combination of the items in the list. Likewise, the term "and/or" in reference to a list of two or more items, covers all of the following interpretations of the word: any one of the items in the list, all of the items in the list, and any combination of the items in the list.

In some embodiments, certain operations, acts, events, or functions of any of the algorithms described herein can be performed in a different sequence, can be added, merged, or left out altogether (e.g., not all are necessary for the practice of the algorithms). In certain embodiments, operations, acts, functions, or events can be performed concurrently, e.g., through multi-threaded processing, interrupt processing, or multiple processors or processor cores or on other parallel architectures, rather than sequentially.

Systems and modules described herein may comprise software, firmware, hardware, or any combination(s) of software, firmware, or hardware suitable for the purposes described. Software and other modules may reside and execute on servers, workstations, personal computers, computerized tablets, PDAs, and other computing devices suitable for the purposes described herein. Software and other modules may be accessible via local computer memory, via a network, via a browser, or via other means suitable for the purposes described herein. Data structures described herein may comprise computer files, variables, programming arrays, programming structures, or any electronic information storage schemes or methods, or any combinations thereof, suitable for the purposes described herein. User interface elements described herein may comprise elements from graphical user interfaces, interactive voice response, command line interfaces, and other suitable interfaces.

Further, processing of the various components of the illustrated systems can be distributed across multiple machines, networks, and other computing resources. Two or more components of a system can be combined into fewer components. Various components of the illustrated systems can be implemented in one or more virtual machines, rather than in dedicated computer hardware systems and/or computing devices. Likewise, the data repositories shown can represent physical and/or logical data storage, including, e.g., storage area networks or other distributed storage systems. Moreover, in some embodiments the connections between the components shown represent possible paths of data flow, rather than actual connections between hardware. While some examples of possible connections are shown, any of the subset of the components shown can communicate with any other subset of components in various implementations.

Embodiments are also described above with reference to flow chart illustrations and/or block diagrams of methods, apparatus (systems) and computer program products. Each block of the flow chart illustrations and/or block diagrams, and combinations of blocks in the flow chart illustrations and/or block diagrams, may be implemented by computer program instructions. Such instructions may be provided to a processor of a general purpose computer, special purpose computer, specially-equipped computer (e.g., comprising a high-performance database server, a graphics subsystem, etc.) or other programmable data processing apparatus to produce a machine, such that the instructions, which execute via the processor(s) of the computer or other programmable data processing apparatus, create means for implementing the acts specified in the flow chart and/or block diagram block or blocks. These computer program instructions may also be stored in a non-transitory computer-readable memory that can direct a computer or other programmable data processing apparatus to operate in a particular manner, such that the instructions stored in the computer-readable memory produce an article of manufacture including instruction means which implement the acts specified in the flow chart and/or block diagram block or blocks. The computer program instructions may also be loaded to a computing device or other programmable data processing apparatus to cause operations to be performed on the computing device or other programmable apparatus to produce a computer implemented process such that the instructions which execute on the computing device or other programmable apparatus provide steps for implementing the acts specified in the flow chart and/or block diagram block or blocks.

Any patents and applications and other references noted above, including any that may be listed in accompanying filing papers, are incorporated herein by reference. Aspects of the invention can be modified, if necessary, to employ the systems, functions, and concepts of the various references described above to provide yet further implementations of the invention. These and other changes can be made to the invention in light of the above Detailed Description. While the above description describes certain examples of the invention, and describes the best mode contemplated, no matter how detailed the above appears in text, the invention can be practiced in many ways. Details of the system may vary considerably in its specific implementation, while still being encompassed by the invention disclosed herein. As noted above, particular terminology used when describing certain features or aspects of the invention should not be taken to imply that the terminology is being redefined herein to be restricted to any specific characteristics, features, or aspects of the invention with which that terminology is associated. In general, the terms used in the following claims should not be construed to limit the invention to the specific examples disclosed in the specification, unless the above Detailed Description section explicitly defines such terms. Accordingly, the actual scope of the invention encompasses not only the disclosed examples, but also all equivalent ways of practicing or implementing the invention under the claims.

To reduce the number of claims, certain aspects of the invention are presented below in certain claim forms, but the applicant contemplates other aspects of the invention in any number of claim forms. For example, while only one aspect of the invention is recited as a means-plus-function claim under 35 U.S.C sec. 112(f) (AIA), other aspects may likewise be embodied as a means-plus-function claim, or in other forms, such as being embodied in a computer-readable medium. Any claims intended to be treated under 35 U.S.C. § 112(f) will begin with the words "means for," but use of the term "for" in any other context is not intended to invoke treatment under 35 U.S.C. § 112(f). Accordingly, the applicant reserves the right to pursue additional claims after filing this application, in either this application or in a continuing application.

What is claimed is:

1. A computer-implemented method comprising:

by a first computing device, generating a list of first assets that are associated with a first application that operates in a first cloud computing environment, wherein the first application is accessed by a first cloud service account, wherein the first computing device comprises one or more hardware processors;

by the first computing device, generating a definition of a first application-entity that corresponds to the first application and that is associated with the list of first assets associated with the first application;

by the first computing device, discovering, in a second cloud computing environment, second assets associated with the first application;

by the first computing device, associating the first application-entity with the second assets;

by the first computing device, initiating a storage management operation for the first application-entity, wherein the storage management operation generates a first set of copies of the first assets and of the second assets associated with the first application, and wherein the first set of copies represent a point-in-time copy of the first application at a time when the storage management operation was initiated for the first application-entity;

by the first computing device, causing the first set of copies to migrate to a third cloud computing environment that is distinct from the first cloud computing environment and from the second cloud computing environment; and

by the first computing device, activating the first application at the third cloud computing environment based on a cloud services definition that is suitable for the third cloud computing environment, wherein in the third cloud computing environment, the first application uses third assets, configured in the third cloud computing environment, that correspond to at least some of the first assets and the second assets; and

wherein migrating the first set of copies to the third cloud computing environment comprises converting an asset mapping associated with the first application-entity into the cloud services definition that is suitable for the third cloud computing environment.

2. The method of claim **1** further comprising:

by the first computing device, causing the first set of copies to be restored as primary data to one or more data storage resources at the third cloud computing environment; and

wherein the third assets comprise at least some of the primary data restored to the one or more data storage resources at the third cloud computing environment.

3. The method of claim **1**, wherein the first assets associated with the first application are discovered by the first computing device based on a cloud services definition in the first cloud service account.

4. The method of claim **1**, further comprising:

by the first computing device, based on determining one or more dependencies among the first assets and the second assets associated with the first application, generating the asset mapping for the first application-entity, wherein the asset mapping comprises the one or more dependencies.

5. The method of claim **1**, further comprising: by the first computing device, based on determining one or more dependencies among the first assets and the second assets associated with the first application, generating the asset mapping for the first application-entity, wherein the asset mapping comprises the one or more dependencies, and wherein the first set of copies are generated according to a first order of operations that is based on the one or more dependencies.

6. The method of claim **1**, wherein the definition of the first application-entity is stored in a database associated with the first computing device, and wherein the database comprises one or more preferences for protecting the first application-entity.

7. The method of claim **1**, wherein the storage management operation comprises at least one of: a full backup of the first application-entity, an incremental backup of the first application-entity, a differential backup of the first application-entity, continuous replication of the first application-entity, and live synchronization of the first application-entity.

8. The method of claim **1**, wherein the first assets and the second assets associated with the first application comprise one or more of:

a virtual computing environment, a template for the virtual computing environment, secure login information, storage for temporary data, persistent storage, metadata, virtual network resources, virtual network configurations, a snapshot, a security group, a configuration file, a lambda function, a native cloud application, a virtual machine, a database, a file storage resource, and a block storage resource.

9. The method of claim **1**, wherein the first computing device comprises a virtual computing resource that executes in one of: the second cloud computing environment and the first cloud computing environment.

10. The method of claim **1**, wherein the first computing device comprises a virtual computing resource that executes in Hall the third cloud computing environment, which is different from the second cloud computing environment and from the first cloud computing environment.

11. The method of claim **1**, wherein the first computing device comprises a virtual computing resource that executes in a different cloud service account from the first cloud service account in the first cloud computing environment.

12. A computer-implemented method comprising:

by a first computing device, generating a list of first assets that are associated with a first application that operates in a first cloud computing environment, wherein the first application is accessed by a first cloud service account, wherein the first assets are discovered by the first computing device based on a cloud services definition in the first cloud service account, and wherein the first computing device comprises one or more hardware processors;

by the first computing device, generating a definition of a first application-entity that corresponds to the first application and that is associated with the list of first assets;

by the first computing device, discovering, in a second cloud computing environment, second assets associated with the first application;

by the first computing device, associating the first application-entity with the second assets;

initiating, by the first computing device, a storage management operation for the first application-entity, wherein the storage management operation generates a first set of copies of the first assets and of the second assets, and wherein the first set of copies represent a point-in-time copy of the first application at a time when the storage management operation was initiated for the first application-entity;

by the first computing device, causing the first set of copies to migrate to a non-cloud data center that is distinct from the first cloud computing environment and from the second cloud computing environment; and

activating the first application at the non-cloud data center, wherein at the non-cloud data center, the first application uses third assets, which are configured from the first assets and from the second assets into configurations that are suitable to the non-cloud data center; and

wherein migrating the first set of copies to the non-cloud data center comprises converting an asset mapping associated with the first application entity into the configurations that are suitable to the non-cloud data center.

13. The method of claim **12** further comprising:

by the first computing device, causing the first set of copies to be restored as primary data to one or more data storage resources at the non-cloud data center; and

wherein the third assets comprise at least some of the primary data restored to the one or more data storage resources at the non-cloud data center.

14. The method of claim **12** further comprising: by the first computing device, based on determining one or more dependencies among the first assets and the second assets, generating the asset mapping for the first application-entity, wherein the asset mapping comprises the one or more dependencies, and wherein the first set of copies are generated according to a first order of operations that is based on the one or more dependencies.

15. The method of claim **12**, wherein the first computing device stores the definition of the first application-entity into a database associated with the first computing device, and wherein the database comprises one or more preferences for protecting the first application-entity.

16. A computer-implemented method comprising:

by a first computing device, discovering first assets associated with a first application that executes in a non-cloud data center;

by the first computing device, generating a definition of a first application-entity that corresponds to the first application and that is associated with the first assets;

initiating by the first computing device, a storage management operation for the first application-entity, wherein the storage management operation generates a first set of copies of the first assets associated with the first application;

by the first computing device, accessing a cloud computing environment, which is distinct from the non-cloud data center;

by the first computing device, generating a cloud services definition for the first application in the cloud computing environment, based at least in part on converting an asset mapping associated with the first application-entity into the cloud services definition, which is suitable for the cloud computing environment;

causing, by the first computing device, the first set of copies to be migrated to the cloud computing environment; and

activating the first application at the cloud computing environment, based at least in part on the cloud services definition, wherein in the cloud computing environment, the first application uses assets migrated from the first set of copies.

17. The method of claim **16**, wherein the first set of copies are stored into storage resources in the non-cloud data center.

18. The method of claim **16** wherein the first computing device is part of a data storage management system, and further comprising: by the first computing device, causing the first set of copies to be restored as primary data to one or more data storage resources at the cloud computing environment; and

wherein the assets used by the first application comprise at least some of the primary data restored to the one or more data storage resources at the cloud computing environment.

19. The method of claim **16** wherein the first computing device is part of a data storage management system, and further comprising: by the first computing device, based on determining one or more dependencies among the first assets, generating the asset mapping for the first application-entity, wherein the asset mapping comprises the one or more dependencies, and wherein the first set of copies are generated according to a first order of operations that is based on the one or more dependencies.

20. The method of claim **16**, wherein the first computing device is part of a data storage management system, wherein the first computing device stores the definition of the first application-entity into a database associated with the first computing device, and wherein the database comprises one or more preferences for protecting the first application-entity.

* * * * *